Android

ListView, Menus, Fragments e Drawers

ListActivity e ListAdapter

```xml
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="match_parent">

    <ListView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/listView"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true" />

</RelativeLayout>
```

```java
public class ListaContrato extends Activity
        implements AdapterView.OnItemClickListener {
    private ListView listView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.list_contato);

        listView = (ListView)findViewById(R.id.listView);
        listView.setAdapter(new DetalheContrato());
        listView.setOnItemClickListener(this);
    }
```

SENAI SÃO PAULO

BaseAdapter

```
public class Item extends BaseAdapter {
    private List<Nome> lista;

    // Declarar um construtor para receber a lista
    // e efetuar a devida inicialização

    @Override
    public int getCount() { return lista.size(); }

    @Override
    public Object getItem(int id) { return lista.get(lista.indexOf(id)); }

    @Override
    public long getItemId(int i) { return lista.get(i).getId(); }

    @Override
    public View getView(int i, View view, ViewGroup viewGroup) {
```

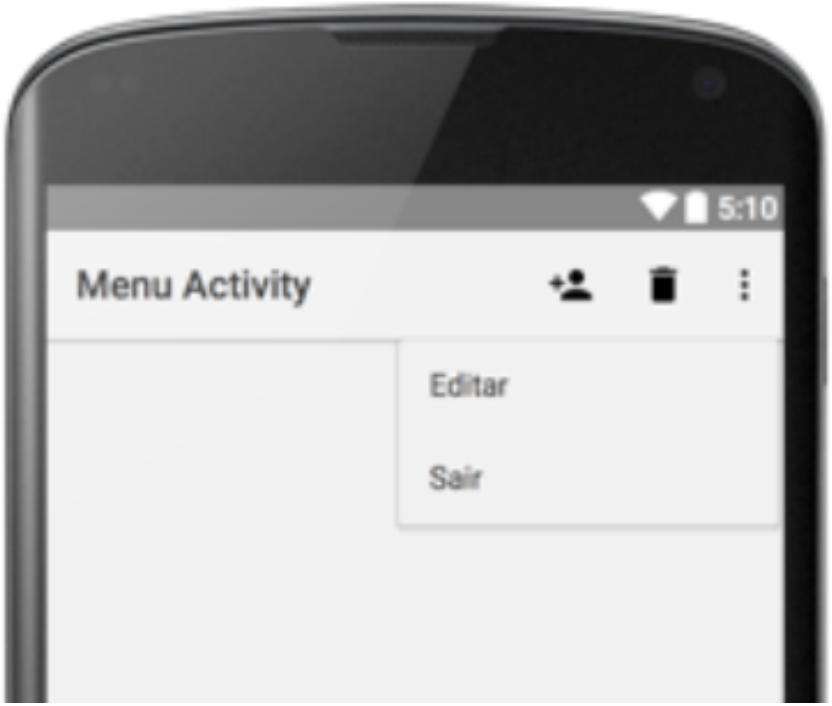
SENAI SÃO PAULO

Chamando um Action

```
@Override
public void onItemClick(AdapterView<?> adapterView, View view, int pos, long id) {
    Intent tela = new Intent(getBaseContext(), EditaContato.class);
    tela.putExtra("id", id);
    startActivityForResult(tela, ACTIVITY_EDITA);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if(requestCode == ACTIVITY_EDITA) {
        if(resultCode == RESULT_OK) {
            ((BaseAdapter)listView.getAdapter()).notifyDataSetChanged();
        }
    }
}
```

Menu

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".MainActivity">
    <item
        android:id="@+id/action_novo"
        android:orderInCategory="100"
        android:icon="@drawable/ic_person_add_black_24dp"
        android:showAsAction="always|withText"
        android:title="@string/action_novo" />
    <item
        android:id="@+id/action_apaga"
        android:orderInCategory="100"
        android:icon="@drawable/ic_delete_black_24dp"
        android:showAsAction="always|withText"
        android:title="@string/action_apaga" />
    <item
        android:id="@+id/action_edita"
        android:orderInCategory="100"
        android:icon="@drawable/ic_create_black_24dp"
        android:showAsAction="ifRoom"
        android:title="@string/action_edita" />
    <item
        android:id="@+id/action_sair"
        android:orderInCategory="100"
        android:icon="@drawable/ic_exit_to_app_black_24dp"
        android:showAsAction="ifRoom"
        android:title="@string/action_sair" />
</menu>
```



Menu

Registro do Menu na Activity

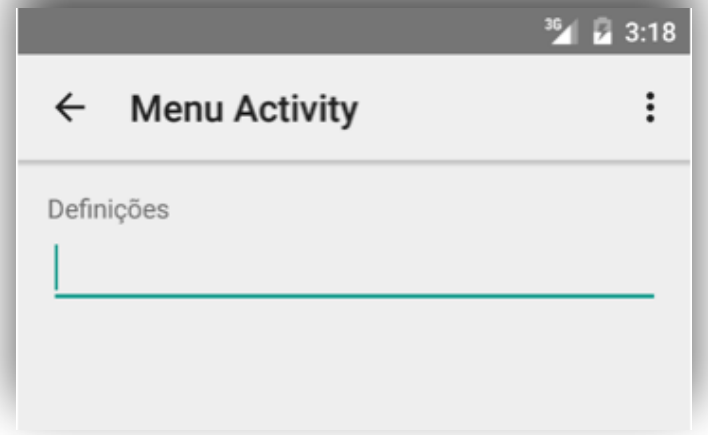
```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_detalhe, menu);
    return true;
}
```

Implementação do OnOptionsItemSelected

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();

    switch (id) {
        case R.id.action_novo:
            // executa a ação
            break;
        case R.id.action_apaga:
            // executa a ação
            break;
        case R.id.action_edita:
            // executa a ação
            break;
        case R.id.action_sair:
            finish();
    }

    return true;
}
```

Action Bar

O Action Bar Home Button Up foi criado com o objetivo da navegação para a Activity parente

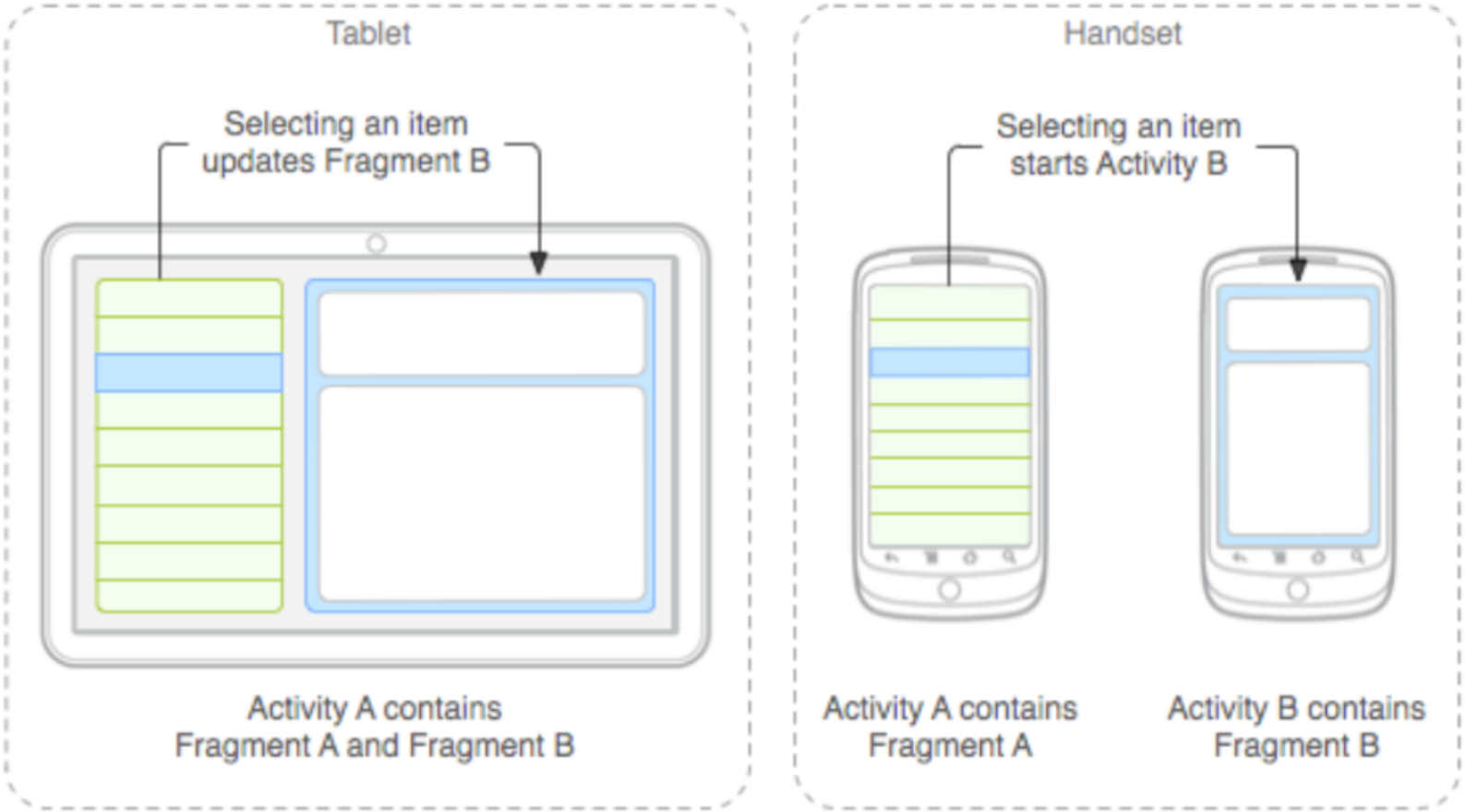



```
ActionBar actionBar = getActionBar();
if(actionBar != null) {
    actionBar.setDisplayHomeAsUpEnabled(true);
    actionBar.setHomeButtonEnabled(true);
}
```

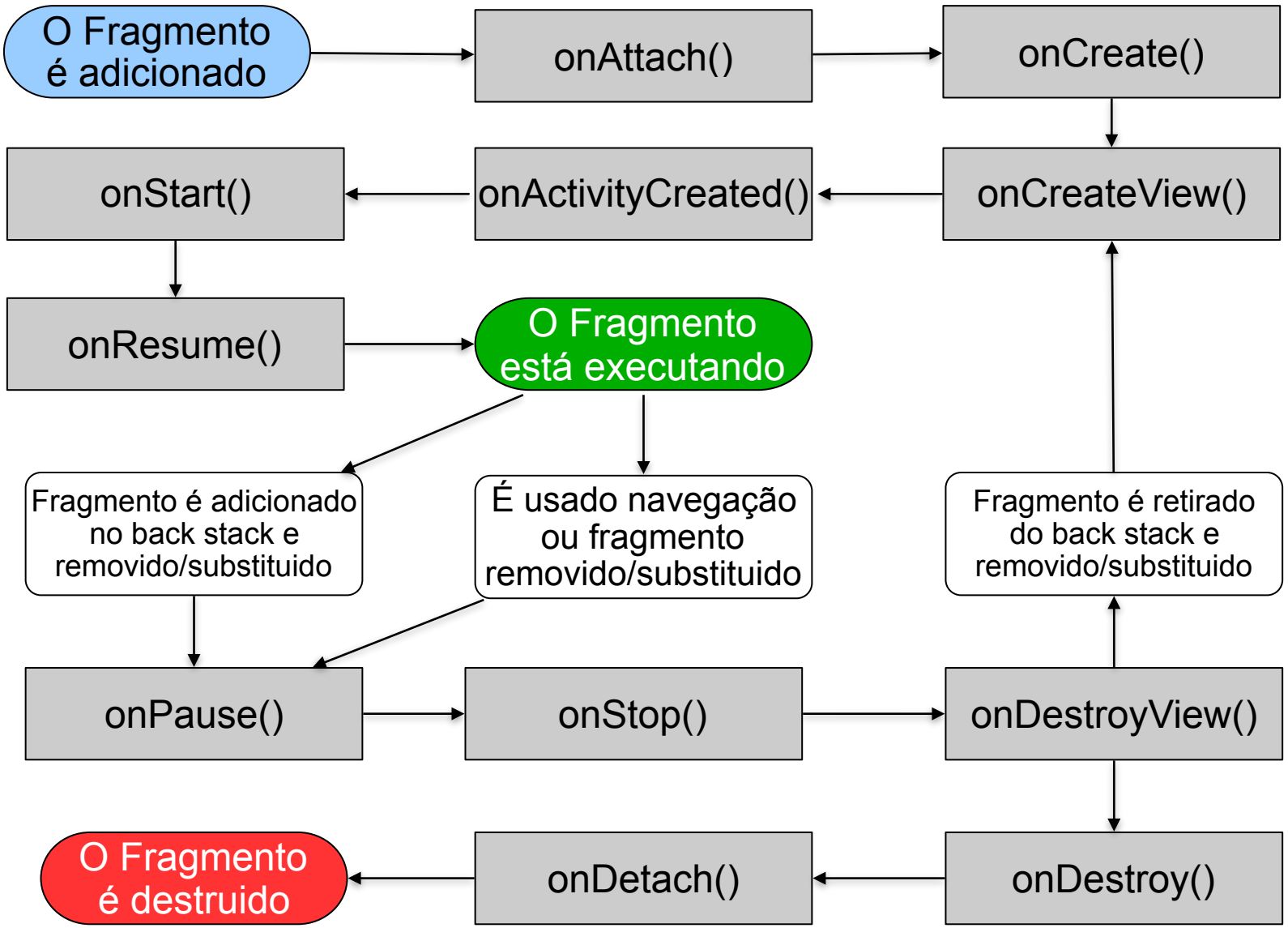
```
<activity android:name=".EditActivity"
    android:parentActivityName=".MainActivity">
    <meta-data android:name="android.support.PARENT_ACTIVITY"
        android:value=".MainActivity" />
</activity>
```

Fragments

Os Fragments permitem a construção de frações de uma Activity possibilitando o seu reuso e também a construção de interfaces complexas.



O Ciclo de vida do Fragment



Fragments

O **Fragment** é definido com um elemento XML **FrameLayout**

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/fragment" />
```

```
@Override
public void onStart() {
    super.onStart();

    ListView listView = (ListView) getActivity().findViewById(R.id.listView);
    listView.setAdapter(new DetalheContrato());
    listView.setOnItemClickListener(this);
}
```

A inicialização de seus componentes deve ficar no método **onStart()**

Sua inicialização se dá através do **FragmentManager**

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.tela_principal);

    if (savedInstanceState == null) {
        getFragmentManager().beginTransaction()
                .replace(R.id.fragment, new ListaContato())
                .commit();
    }
}
```

DatePickerDialog

O **DatePickerDialog** é utilizado para a edição de Datas, porém para sua utilização é necessário a criação de um **DialogFragment** e a implementação do método **onDateSet** da interface **OnDateSetListener**.

```
@Override
public void onClick(View view) {
    DialogFragment fragment = new DialogFragment() {
        @Override
        public Dialog onCreateDialog(Bundle savedInstanceState) {
            DatePickerDialog.OnDateSetListener listener = new DatePickerDialog.OnDateSetListener() {
                @Override
                public void onDateSet(DatePicker view, int ano, int mes, int dia) {
                    dataNascimento.set(ano, mes, dia);
                    edData.setText(fmt.format(dataNascimento.getTime()));
                }
            };

            try {
                dataNascimento.setTime(fmt.parse(edData.getText().toString()));
            } catch (ParseException ex) {}

            int dia = dataNascimento.get(Calendar.DAY_OF_MONTH);
            int mes = dataNascimento.get(Calendar.MONTH);
            int ano = dataNascimento.get(Calendar.YEAR);

            DatePickerDialog dialog =  new DatePickerDialog(getActivity(), listener, ano, mes, dia);

            return dialog;
        }
    };
    fragment.show(getFragmentManager(), "Data de Nascimento");
}
```

DatePickerDialog

Na API 26 passa a ser necessário a implementação do DialogFragment em uma classe própria e que todos os atributos que são utilizados pelo DatePickerDialog devem ser passados via métodos setter.

```
public class DateDialog extends DialogFragment implements DatePickerDialog.OnDateSetListener {
    private Calendar calendar;
    private EditText editText;
    private static DateFormat fmt =  DateFormat.getDateInstance(DateFormat.LONG);

    public static DateDialog makeDialog(Calendar calendar, EditText editText) {
        DateDialog dialog = new DateDialog();
        dialog.calendar = calendar;
        dialog.editText = editText;
        return dialog;
    }

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        if(calendar == null) {
            long cal = savedInstanceState.getLong( key: "cal");
            calendar = Calendar.getInstance();
            calendar.setTimeInMillis(cal);
        }

        int dia = calendar.get(Calendar.DAY_OF_MONTH);
        int mes = calendar.get(Calendar.MONTH);
        int ano = calendar.get(Calendar.YEAR);

        DatePickerDialog dialog = new DatePickerDialog(getActivity(), listener: this, ano, mes, dia);
        return dialog;
    }

    @Override
    public void onDateSet(DatePicker view, int ano, int mes, int dia) {
        calendar.set(ano, mes, dia);
        editText.setText(fmt.format(calendar.getTime()));
    }

    @Override
    public void onSaveInstanceState(Bundle outState) {
        outState.putLong("cal", calendar.getTimeInMillis());
        super.onSaveInstanceState(outState);
    }
}
```

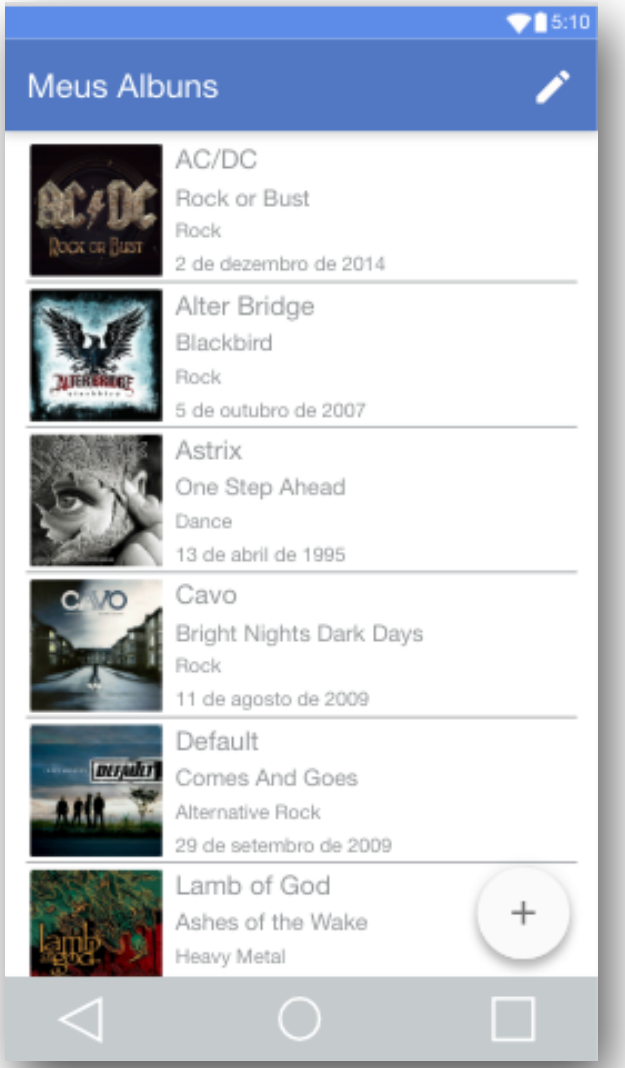
DatePickerDialog

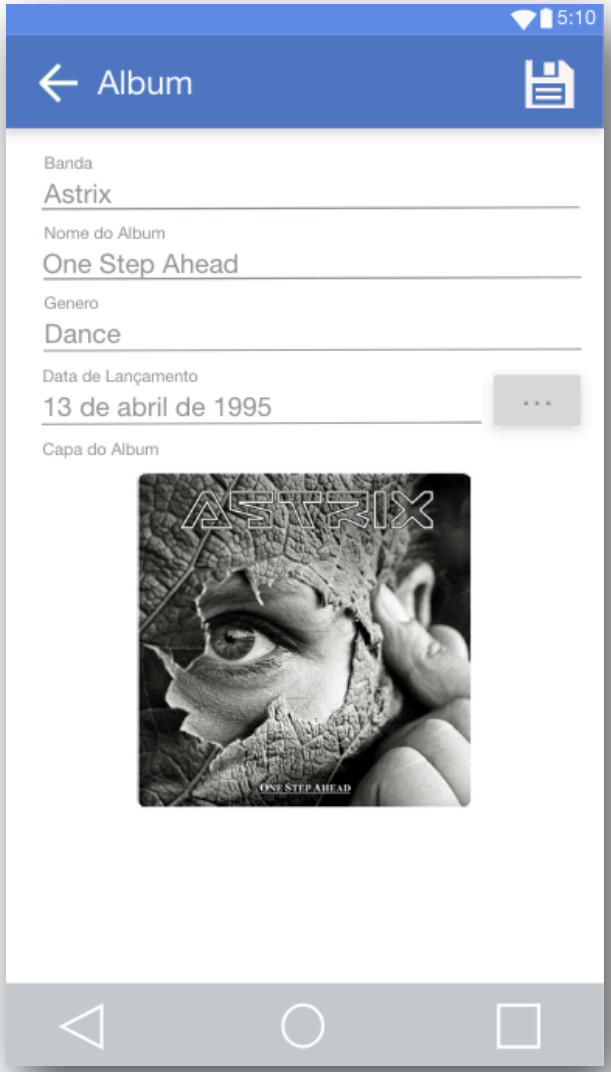
Para a utilização da classe onde o DialogFragment foi declarado será necessário informar todos os valores necessários, caso contrario ocorrerá uma Exception.

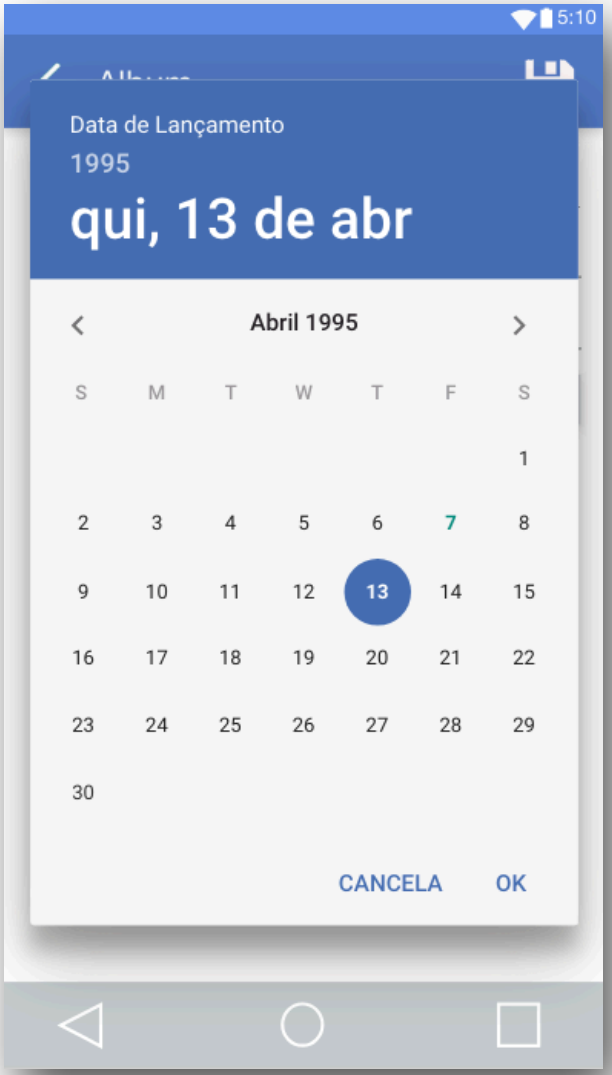
```
@Override
public void onClick(View view) {
    if(view.equals(btLancamento)) {
        // Abrir o Date PickerDialog
        selecionaData(view);
    } else {
        // Abrir a Galeria de Fotos
        abrirGalery();
    }
}

public void selecionaData(View view) {
    DateDialog.makeDialog(calendar,edLancamento)
            .show(getFragmentManager(), tag: "Data de Lançamento");
}
```

SENAI SÃO PAULO

Meus Albuns



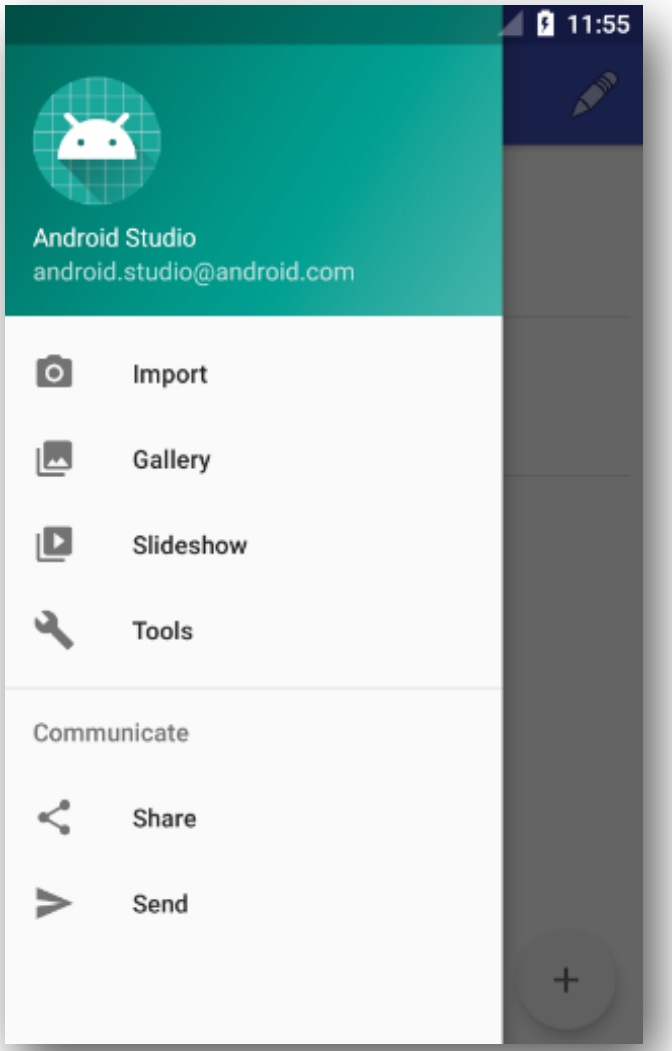




Navigation Drawer

O Menu Lateral é construído com o DrawerLayout. Ele é composto pelo Menu de Navegação e a Página Principal.

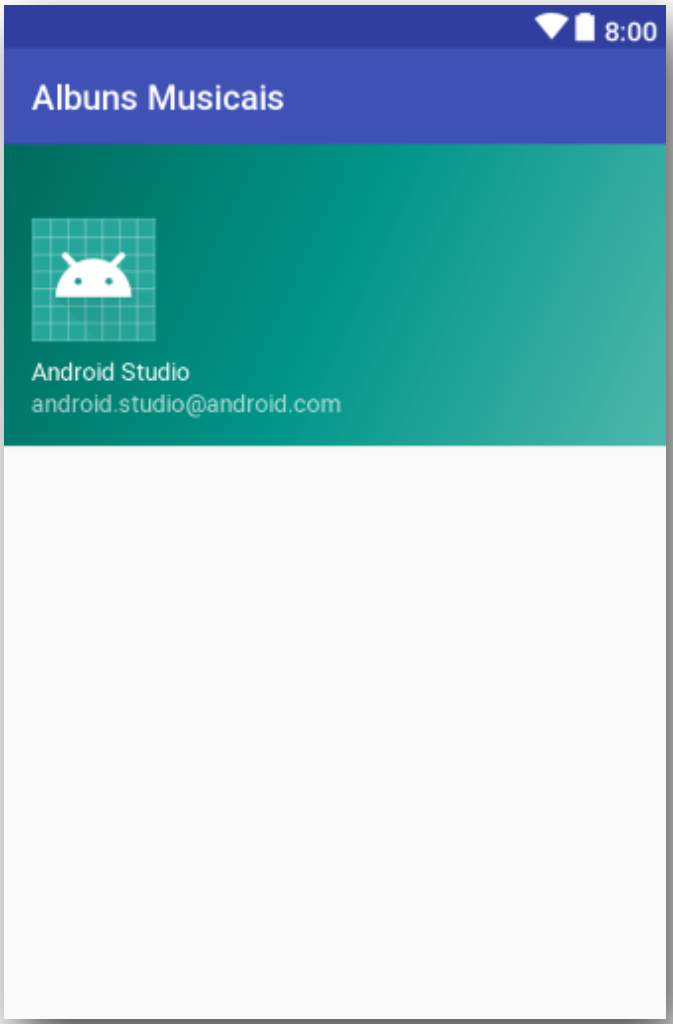
```
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

    <include
        layout="@layout/app_bar_menu"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <android.support.design.widget.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header_menu"
        app:menu="@menu/activity_menu_drawer" />

</android.support.v4.widget.DrawerLayout>
```



Navigation Drawer

O Menu de Navegação é composto pelo cabeçálho. Nele podemos configurar a imagem que desejarmos e textos informativos.

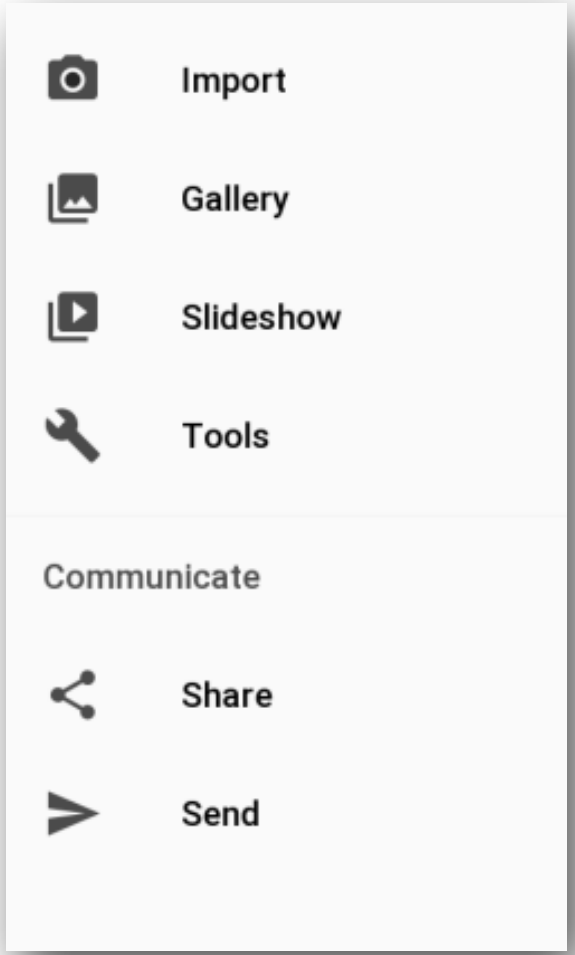
```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="176dp"
    android:background="@drawable/side_nav_bar"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:ignore="HardcodedText, ContentDescription"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingTop="8dp"
        app:srcCompat="@mipmap/ic_launcher_round" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="8dp"
        android:text="Android Studio"
        android:textAppearance="@style/TextAppearance.AppCompat.Body1" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="android.studio@android.com" />

</LinearLayout>
```



Navigation Drawer

As opções do Menu de Navegação são construídas da mesma forma que um menu de aplicação tendo como diferença o atributo **showIn**

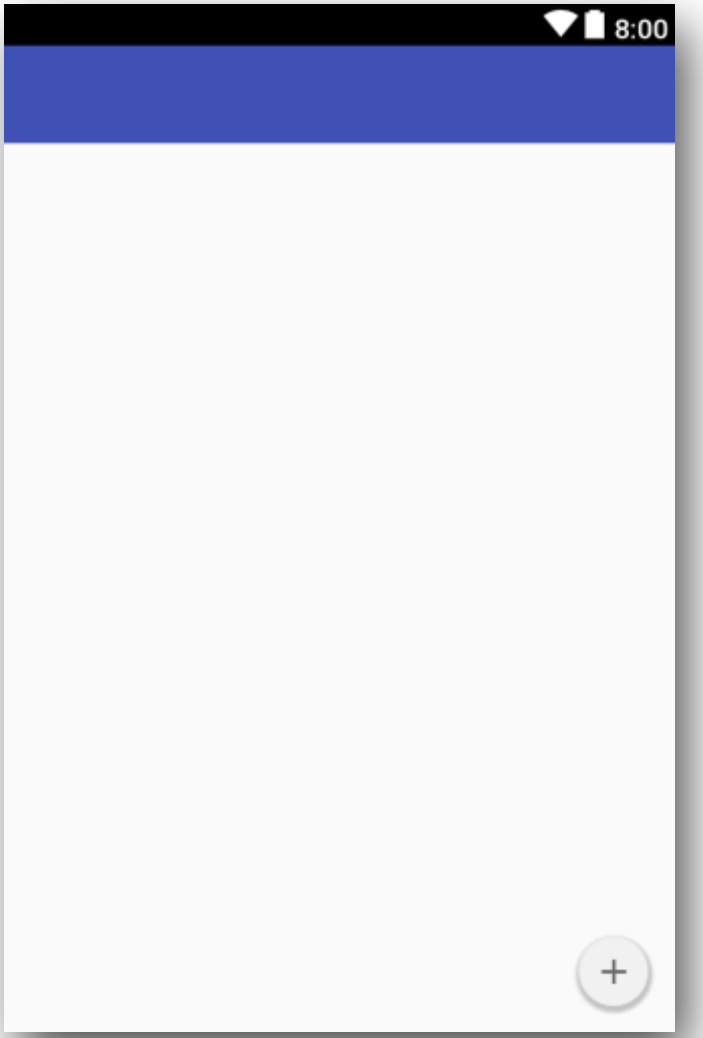
```
<menu
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:showIn="navigation_view"
    tools:ignore="HardcodedText">

    <group android:checkableBehavior="single">
        <item
            android:id="@+id/nav_camera"
            android:icon="@drawable/ic_menu_camera"
            android:title="Import" />
        <item
            android:id="@+id/nav_gallery"
            android:icon="@drawable/ic_menu_gallery"
            android:title="Gallery" />
        <item
            android:id="@+id/nav_slideshow"
            android:icon="@drawable/ic_menu_slideshow"
            android:title="Slideshow" />
        <item
            android:id="@+id/nav_manage"
            android:icon="@drawable/ic_menu_manage"
            android:title="Tools" />
    </group>

    <item android:title="Communicate">
        <menu>
            <item
                android:id="@+id/nav_share"
                android:icon="@drawable/ic_menu_share"
                android:title="Share" />
            <item
                android:id="@+id/nav_send"
                android:icon="@drawable/ic_menu_send"
                android:title="Send" />
        </menu>
    </item>

</menu>
```



Navigation Drawer

A área da Página principal é controlada pelo **CoordinatorLayout** para que seja possível ocultar o **ToolBar** quando o menu de navegação aparecer. Nele coexistem o **AppBarLayout** a **Página Principal da aplicação** e se houver necessidade o **FloatActionButton** que normalmente seria configurado na Página Principal.

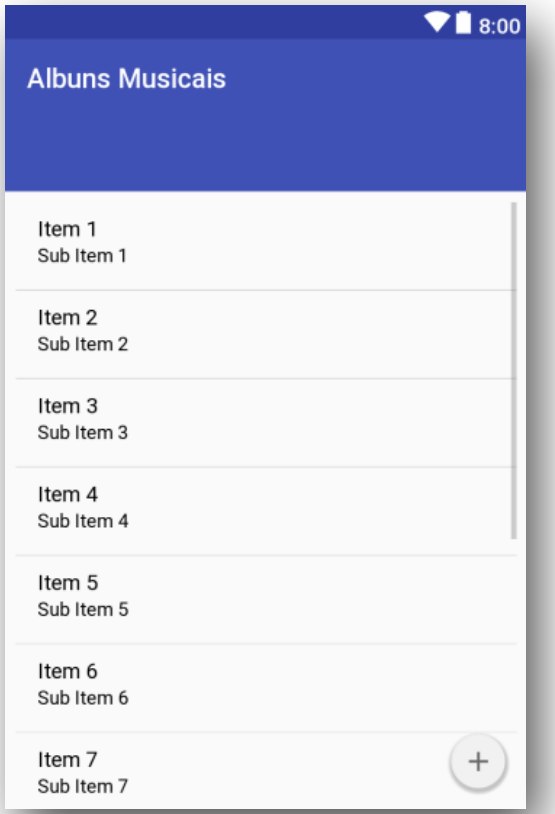
```
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="br.senai.sp.informatica.albunsmusicais.view.MainActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/activity_lista" />

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/btAdd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:backgroundTint="@color/fundoCinza"
        android:layout_gravity="bottom|end"
        android:layout_margin="16dp"
        app:elevation="6dp"
        app:srcCompat="@drawable/ic_add_black_24dp" />
</android.support.design.widget.CoordinatorLayout>
```



Navigation Drawer

A Página principal necessita receber dois atributos para que se encaixe no Menu de Navegação. São eles o app:layout_behavior="@string/appbar_scrolling_view_behavior" e tools:showIn="@layout/app_bar_menu" .

```
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="br.senai.sp.informatica.albunsmusicais.view.MainActivity"
    app:layout_behavior="android.support.design.widget.AppBarLayout$ScrollingVie..."
    tools:showIn="@layout/app_bar_menu">

    <ListView
        android:id="@+id/listView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```



Navigation Drawer

A **MainActivity** deve registrar o novo **layout** e inicializar o **ToolBar**, o **DrawerLayout** e o **NavigationView** no método **onCreate**.

1

2

3

4

```
private DrawerLayout drawer;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_menu);

    FloatingActionButton fab = findViewById(R.id.btAdd);
    fab.setOnClickListener(this);

    albumAdapter = new AlbumAdapter( activity: this);

    ListView listView = findViewById(R.id.listView);
    listView.setAdapter(albumAdapter);
    listView.setOnItemClickListener(this);

    // Configuração do ToolBar
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    // Configuração no Menu de Navegação
    drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            activity: this, drawer, toolbar, "Open navigation drawer", "Close navigation drawer");
    drawer.addDrawerListener(toggle);
    toggle.syncState();

    // Registro dos menu para tratar as ações do menu de navegação
    NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
    navigationView.setNavigationItemSelectedListener(this);
}
```

Navigation Drawer

A **MainActivity** deve implementar a interface **OnNavigationItemSelectedListener** para receber as solicitações de ação do menu lateral. Também deve ser implementado o método **onBackPressed** para que o menu lateral seja recolhido quando a opção Voltar do Android seja selecionada.

1

```
public class MainActivity extends AppCompatActivity
        implements AdapterView.OnItemClickListener, View.OnClickListener,
        NavigationView.OnNavigationItemSelectedListener {
```

2

```
// Método que oculta o Menu de Navegação ao ser selecionada a ação de "Voltar (Back)" do Android
@Override
public void onBackPressed() {
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    }
    super.onBackPressed();
}
```

3

```
// Método que trata as ações do menu de navegação
@Override
public boolean onNavigationItemSelected(@NonNull MenuItem item) {
    int id = item.getItemId();

    switch (id) {
        case R.id.action_settings:
            break;
        default:
            break;
    }

    return true;
}
```

SENAI SÃO PAULO

Meus Albuns