# 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) {
```

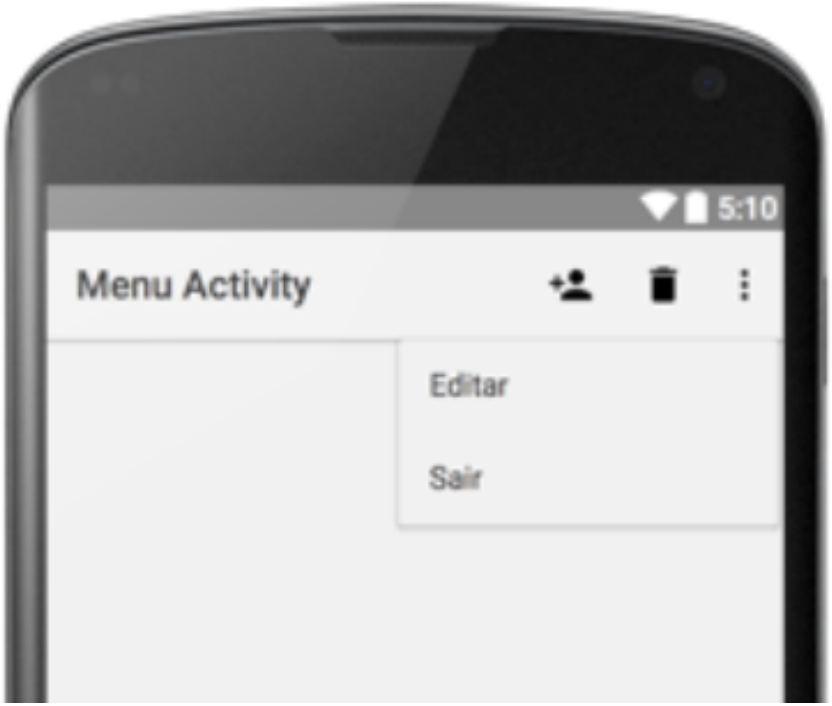
SENAI SÃO PAULO

# 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

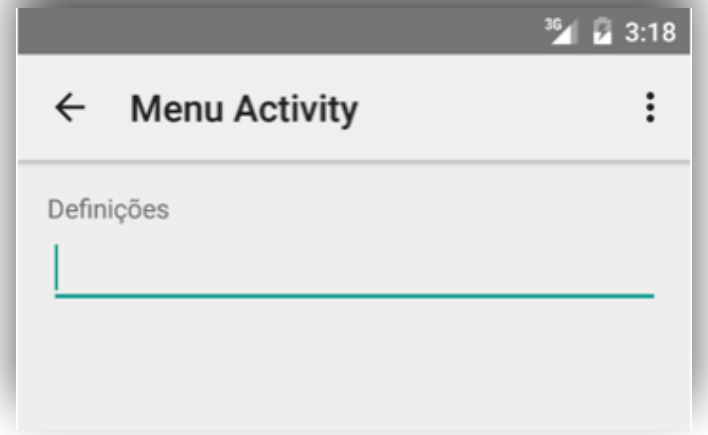
```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_detalhe, menu);
    return true;
}
```

## 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

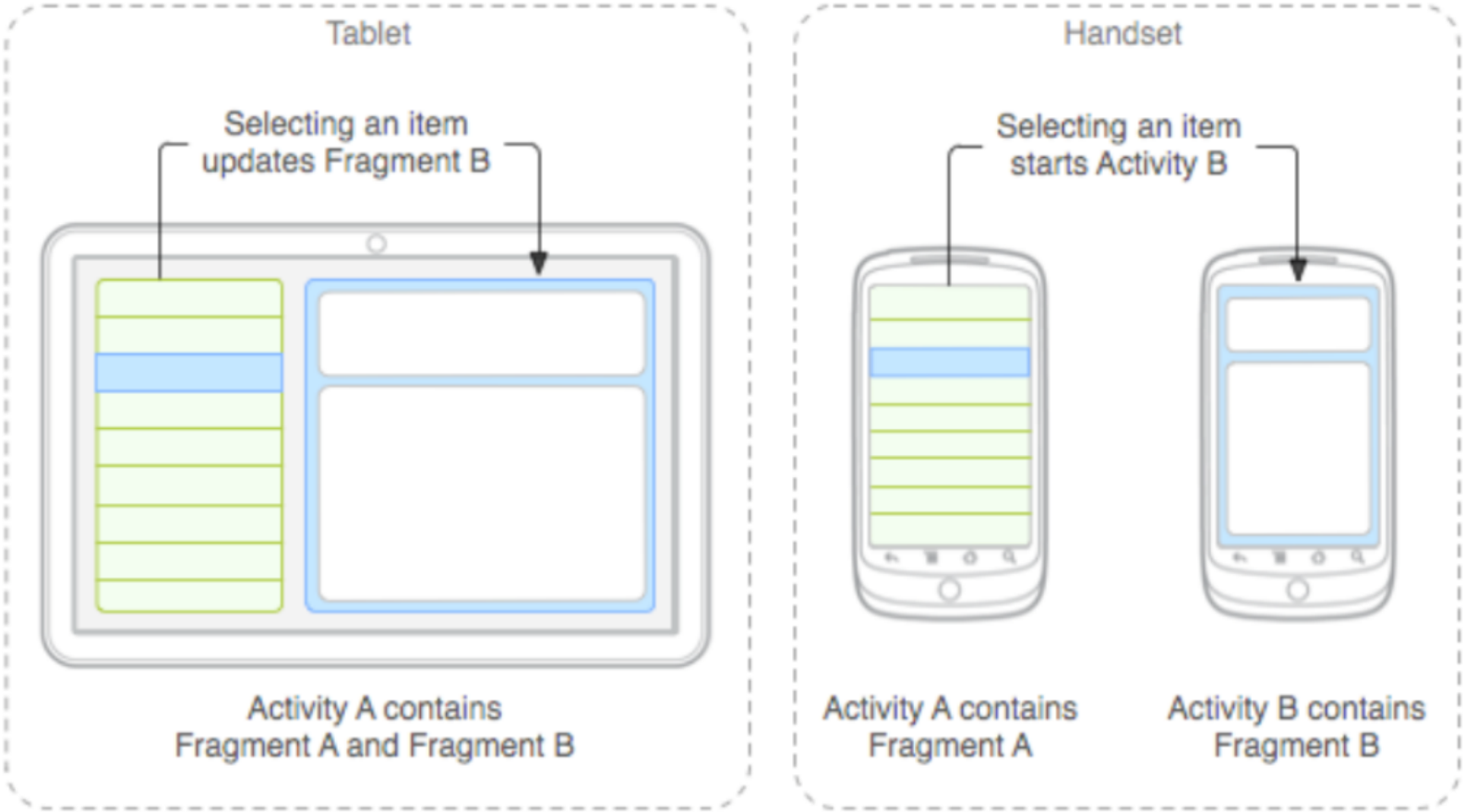



```
ActionBar actionBar = getActionBar();
if(actionBar != null) {
    actionBar.setDisplayHomeAsUpEnabled(true);
    actionBar.setHomeButtonEnabled(true);
}
```

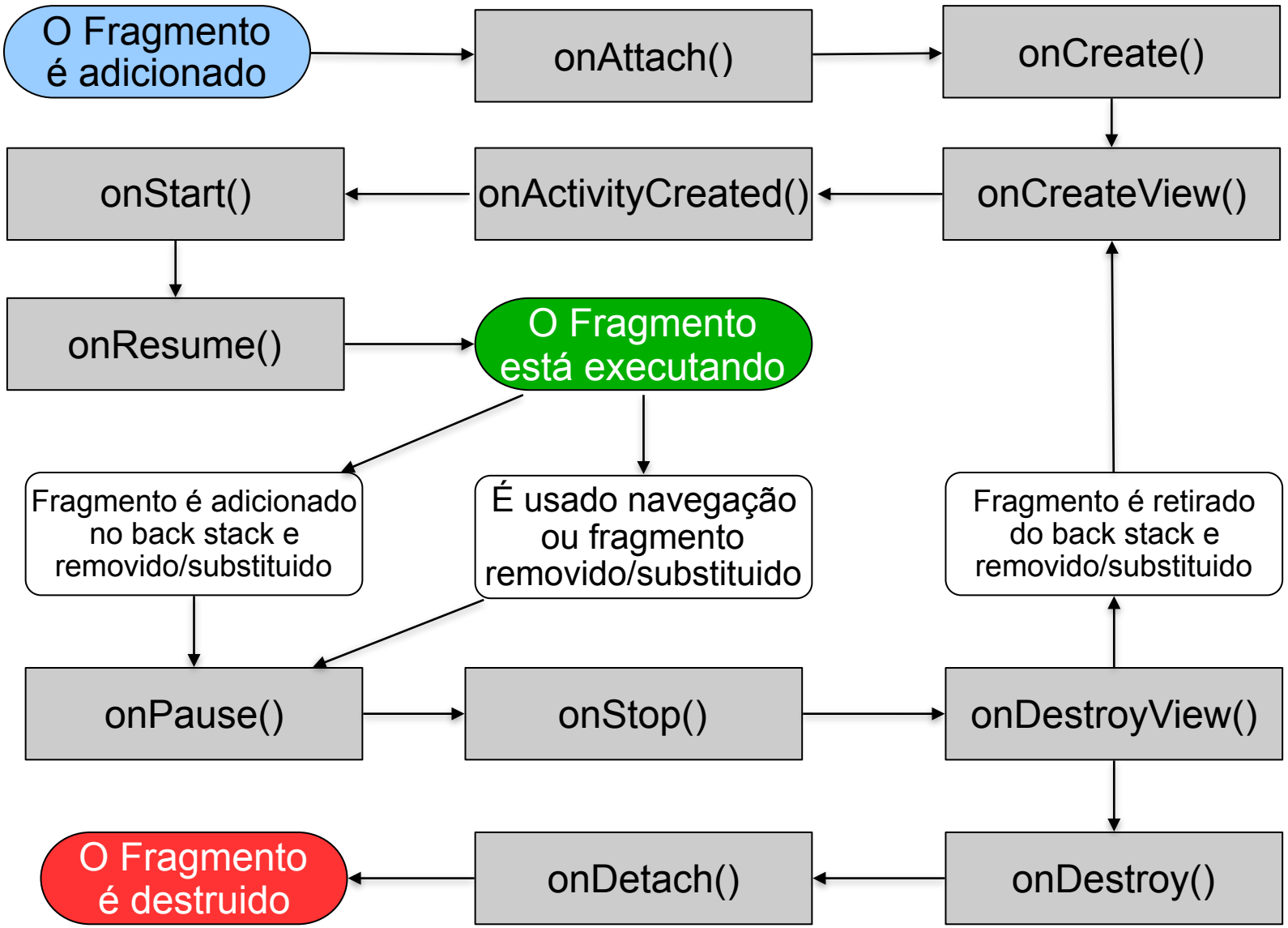
```
<activity android:name=".EditActivity"
    android:parentActivityName=".MainActivity">
    <meta-data android:name="android.support.PARENT_ACTIVITY"
        android:value=".MainActivity" />
</activity>
```

# 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.

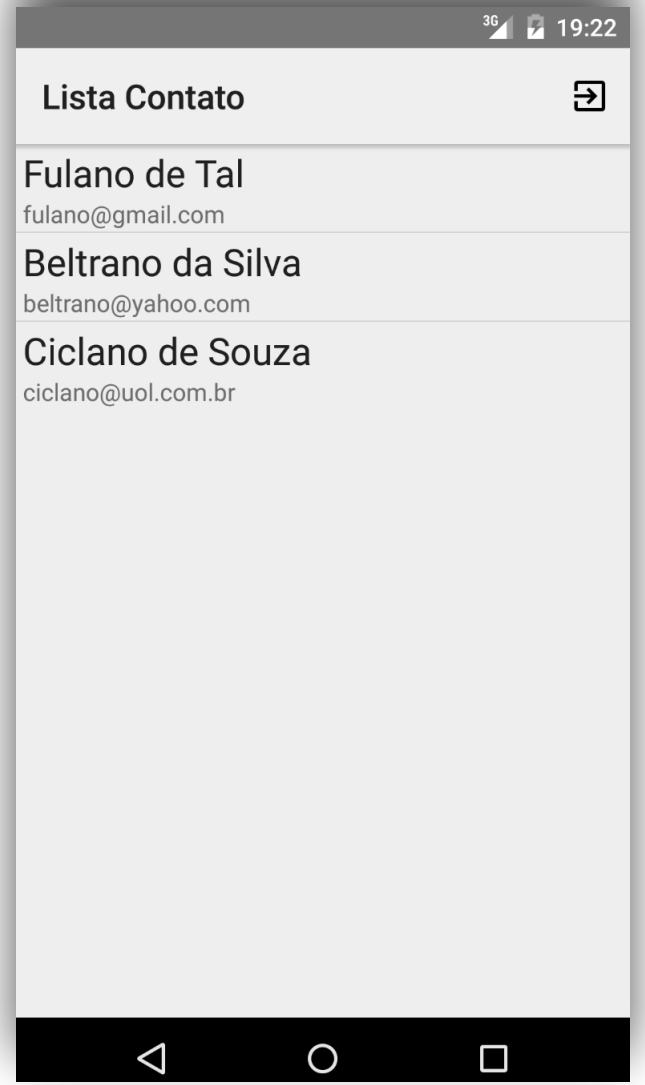
```
public class DateDialog extends DialogFragment {
   private View view;
   private Calendar calendar;
   private EditText editText;
   private static DateFormat fmt =  DateFormat.getDateInstance(DateFormat.LONG);

   @Override
   public Dialog onCreateDialog(Bundle savedInstanceState) {
       DatePickerDialog.OnDateSetListener listener = new DatePickerDialog.OnDateSetListener() {
           @Override
           public void onDateSet(DatePicker view, int ano, int mes, int dia) {
               calendar.set(ano, mes, dia);
               editText.setText(fmt.format(calendar.getTime()));
           }
       };

       try {
           calendar.setTime(fmt.parse(editText.getText().toString()));
       } catch (ParseException ex) {
       }

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

       DatePickerDialog dialog = new DatePickerDialog(view.getContext(), listener, ano, mes, dia);
       return dialog;
   }

   public void setView(View view) { this.view = view; }

   public void setCalendar(Calendar calendar) { this.calendar = calendar; }

   public void setEditText(EditText editText) { this.editText = editText; }
}
```

# 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 dialog = new DateDialog();
    dialog.setView(view);
    dialog.setCalendar(calendar);
    dialog.setEditText(edLancamento);
    dialog.show(getFragmentManager(), tag: "Data de Lançamento");
}
```

# Lista de Contatos

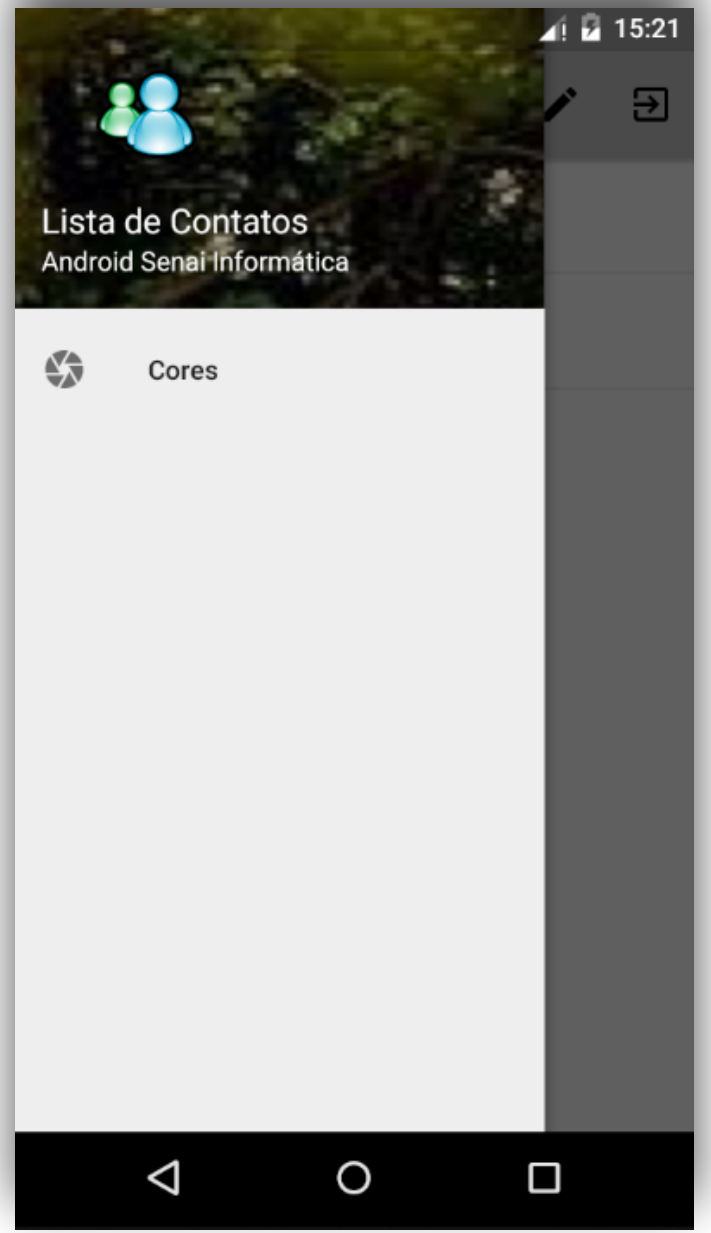




# Navigation Drawer

O **Navigation Drawer** é a composição de vários componentes, tais como o **AppBarLayout** e o **ToolBar.**

```
<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"
    android:fitsSystemWindows="true"
    tools:context="br.senai.sp.cfp132.contatos.view.Principal">

    <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>

    <FrameLayout
        android:id="@+id/fragment"
        android:layout_marginTop="56dp"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</android.support.design.widget.CoordinatorLayout>
```



# Navigation Drawer

A inicialização do **Navigation Drawer** acontece no método **onCreate** da **Activity** principal.

```
public class Principal extends AppCompatActivity
        implements NavigationView.OnNavigationItemSelectedListener {

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

        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
                this, drawer, toolbar, R.string.navigation_drawer_open,
                R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.syncState();

        NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);

        FragmentTransaction tx = getFragmentManager().beginTransaction();
        tx.replace(R.id.fragment, new MainFragment());
        tx.addToBackStack(null);
        tx.commit();

    }
```

# Navigation Drawer

O **DrawerLayout** é utilizado para compor o cabeçalho e o corpo do Drawer.

```
<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_principal"
        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_principal"
        app:menu="@menu/activity_principal_drawer" />

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

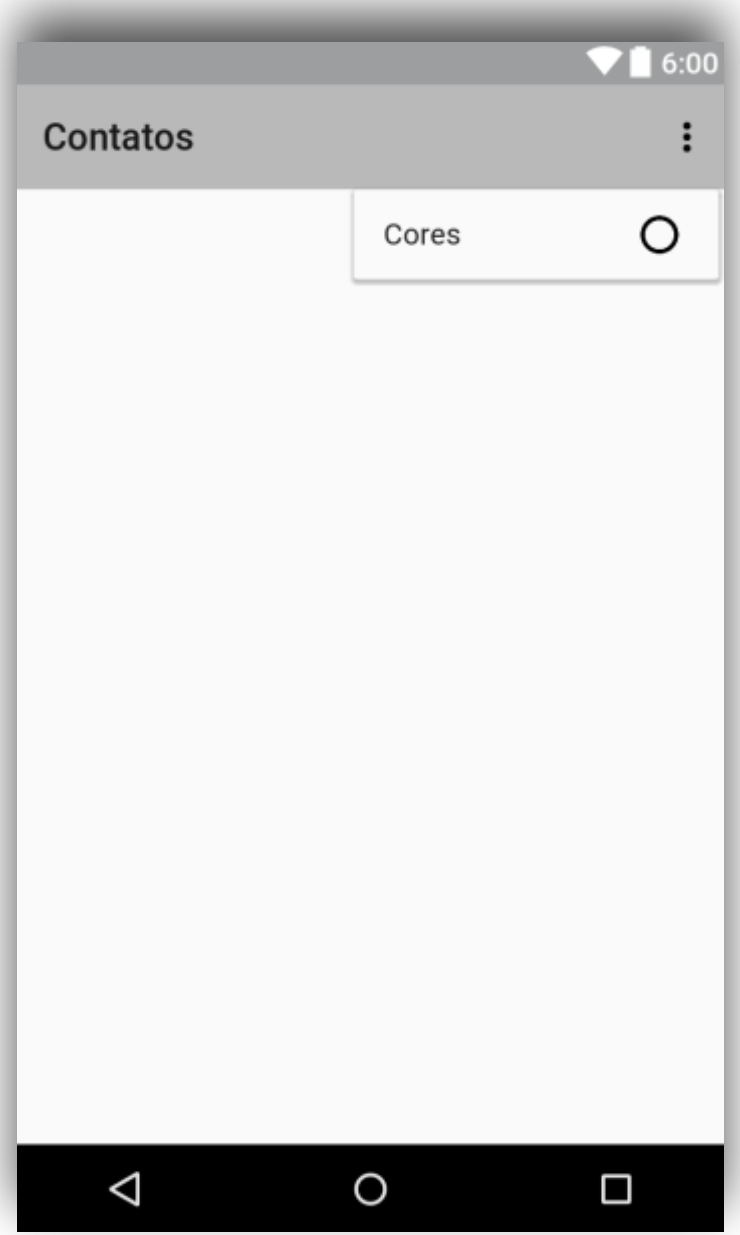
O **Navigation Header** é utilizado para criar o cabeçalho do Drawer.

```
<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="160dp"
    android:background="@drawable/side_nav_bar"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingBottom="16dp"
    android:paddingLeft="64dp"
    android:paddingRight="64dp"
    android:paddingTop="16dp"
    android:theme="@style/ThemeOverlay.AppCompat.Dark"
    android:weightSum="1"
    tools:ignore="ContentDescription" >

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="110dp"
        android:layout_height="70dp"
        android:paddingTop="16dp"
        app:srcCompat="@drawable/nav_bar_icon"/>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="16dp"
        android:text="Lista de Contatos"
        android:textAppearance="@style/TextAppearance.AppCompat.Body1"
        android:textColor="@android:color/background_light"
        android:textSize="18sp"
        android:layout_weight="0.36" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/android_senai_inform_tica"
        android:textColor="@android:color/background_light"
        android:textSize="14sp"
        android:layout_weight="0.36" />

</LinearLayout>
```

# Navigation Drawer

As opções do **Navigation Drawer** são construídas a partir de um **Menu**.

```
<menu
    xmlns:android="http://schemas.android.com/apk/res/android">

    <group android:checkableBehavior="single">
        <item
            android:id="@+id/cores"
            android:icon="@drawable/ic_camera_black_24dp"
            android:title="@string/cores" />
    </group>

</menu>
```



# Lista de Contatos