INSTITUTO FEDERAL
PARANÁ

DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS

PROFª. M.Sc. JULIANA H Q BENACCHIO

Action Bar

INSTITUTO FEDERAL
PARANÁ

- A **action bar** é a barra superior da aplicação
- Surgiu no Android 3.0 (API Level 11)
- Benefícios
  - Identifica a aplicação
  - Permite a execução de ações
  - Torna a navegação mais intuitiva

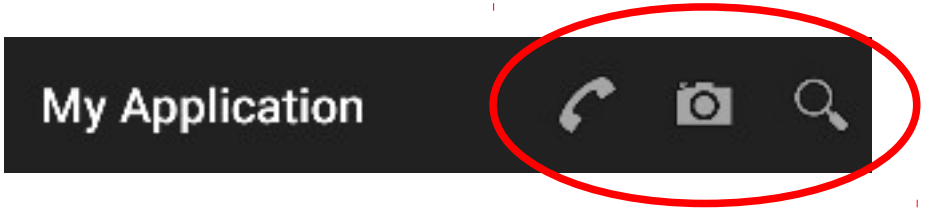
A Classe `ActionBar`

- O objeto da classe **`ActionBar`** é o ponto de partida para gerenciar a action bar via programação
- A instância pode ser obtida através de uma **activity**
  - **`ActionBar actionBar = getActionBar();`**

Action Buttons

INSTITUTO FEDERAL
PARANÁ

- A **action bar** possui suporte aos chamados ***action buttons***



- As ações são normalmente criadas através de um ***resource*** do tipo menu

Action Buttons

INSTITUTO FEDERAL
PARANÁ

/res/menu/activity_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/action_call"
        android:icon="@android:drawable/ic_menu_call"
        app:showAsAction="ifRoom|withText"
        android:title="@string/call" />
    <item
        android:id="@+id/action_camera"
        android:icon="@android:drawable/ic_menu_camera"
        app:showAsAction="ifRoom|withText"
        android:title="@string/camera" />
    <item
        android:id="@+id/action_search"
        android:icon="@android:drawable/ic_menu_search"
        app:showAsAction="ifRoom|withText"
        android:title="@string/search" />
</menu>
```

Action Buttons

INSTITUTO FEDERAL
PARANÁ

- O Android chama **`onCreateOptionsMenu()`** quando é iniciada, para exibir as ações

```
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_activity, menu);
    return super.onCreateOptionsMenu(menu);
}
```

O método inflate() extrai os itens definidos no XML

Deve ser true, mas é uma boa prática chamar o método da superclasse

Action Buttons

INSTITUTO FEDERAL
PARANÁ

- Quando um **action button** é clicado, o método **`onOptionsItemSelected()`** é chamado na **activity**

```
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.action_call) {
        //...
        return true;
    } else if (id == R.id.action_camera) {
        //...
        return true;
    } else if (id == R.id.action_network) {
        //...
        return true;
    } else {
        return super.onOptionsItemSelected(item);
    }
}
```

item.getItemId() retorna o ID do resource clicado

Retorna true se o evento for tratado, ou chama o método da superclasse

Action Buttons

INSTITUTO FEDERAL
PARANÁ

- É possível também definir o atributo **`onClick`** no XML, que referencia o método que será chamado quando o **action button** for clicado

```
<menu
    <item
    ...
        android:onClick="add"
    ... />
```

```
public void add(MenuItem item) {
    //...
}
```

Recebe como parâmetro o item clicado

Visibilidade dos Action Buttons

INSTITUTO FEDERAL
PARANÁ

- O Android determina como os botões serão dispostos na **action bar**
  - De acordo com o tamanho da tela
  - De acordo com parâmetros de configuração
- A configuração pode ser feita na definição do item, no XML
  - Atributo **`showAsAction`**

Visibilidade dos Action Buttons

INSTITUTO FEDERAL
PARANÁ

- Valores do atributo **`showAsAction`**

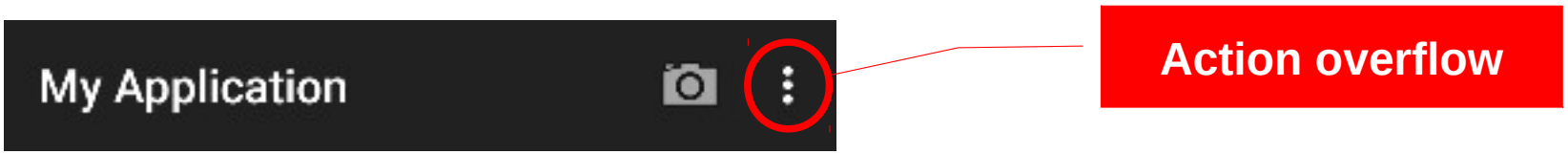
Valor	Comportamento
`ifRoom`	Aparece se houver espaço
`always`	Sempre aparece
`never`	Nunca aparece
`withText`	Aparece junto com seu texto

É possível combinar valores. withText
Ex: ifRoom|withText

Action Overflow

INSTITUTO FEDERAL
PARANÁ

- Local onde os **action buttons** ficam
  - Se marcados como **`showAsAction = never`**
  - Se não há espaço na **action bar**
- A forma de acessar o **action overflow** varia de acordo com o dispositivo

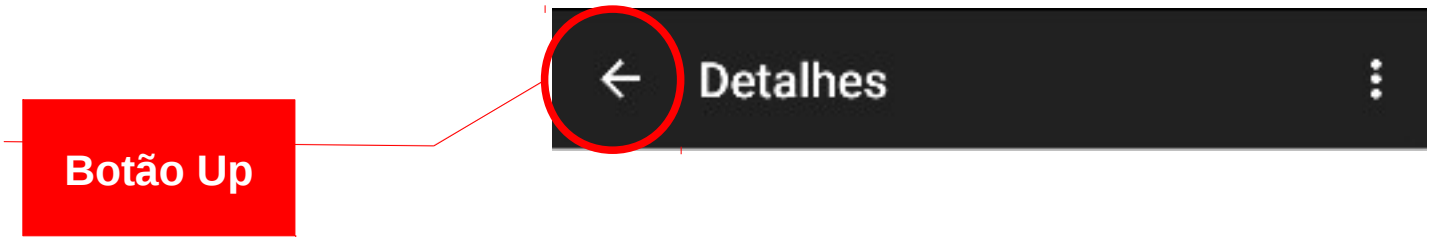


Up Navigation

INSTITUTO FEDERAL
PARANÁ

- A **action bar** permite o que chamamos de **up navigation**
  - Retornar para a tela anterior na hierarquia
- É preciso habilitar na **activity**

```
getActionBar().setDisplayHomeAsUpEnabled(true);
```



Up Navigation

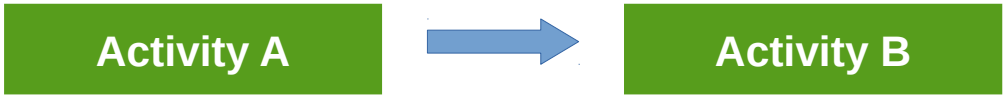
INSTITUTO FEDERAL
PARANÁ

- Apenas habilitar a **up navigation** na **activity** não é o suficiente
  - É preciso determinar para onde ir quando o botão **up** é pressionado
- Existem duas formas de fazer isso
  - Via configuração
  - Via programação

Up Navigation

INSTITUTO FEDERAL
PARANÁ

- A "**activity-pai**" na hierarquia é definida no **`AndroidManifest.xml`**
- Utilizada quando a hierarquia é fixa




```
<activity android:name="ifpr.professor.ActivityA" />
```

```
<activity
   android:name="ifpr.professor.ActivityB"
   android:parentActivityName="ifpr.professor.ActivityA">
   <meta-data
      android:name="android.support.PARENT_ACTIVITY"
      android:value="ifpr.professor.ActivityA" />
</activity>
```

API Level >= 16

API Level < 16

Up Navigation

INSTITUTO FEDERAL
PARANÁ

- O método **`onOptionsItemSelected()`** implementa o comportamento
- Utilizada quando a hierarquia pode variar

```
public boolean onOptionsItemSelected(MenuItem item) {

 if (item.getItemId() == android.R.id.home) {
    Intent i = new Intent(getApplicationContext(), ActivityA.class);
    i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    startActivity(i);
    return true;
 } else {
    return super.onOptionsItemSelected(item);
 }
}
```

Action Bar

INSTITUTO FEDERAL
PARANÁ

- O objeto **`ActionBar`** tem mais alguns métodos interessantes

Método	Descrição
`show()`	Exibe a action bar
`hide()`	Esconde a action bar
`setDisplayShowTitleEnabled()`	Mostra ou esconde o título da activity
`setDisplayShowHomeEnabled()`	Mostra ou esconde o ícone da action bar

- A **action bar** também é totalmente customizável em sua parte gráfica

Action Bar

INSTITUTO FEDERAL
PARANÁ

- A **action bar** surgiu no Android 3.0 (API Level 11)
- Para ser usada no Android 2.1 (API Level 7) ou acima, é preciso usar a **API de compatibilidade**
- Não existe suporte à **action bar** para versões anteriores à 2.1
  - A não ser com o uso de APIs de terceiros

Action Bar

INSTITUTO FEDERAL
PARANÁ

- A **activity** deve herdar de **`AppCompatActivity`** e chamar o método **`getSupportActionBar()`**

```
public class MainActivity extends AppCompatActivity {

   public void onCreate(Bundle savedInstanceState) {
      ...
      ActionBar actionBar = getSupportActionBar();
      ...
   }
}
```

Classe do pacote android.support.v7.app

Action Bar

INSTITUTO FEDERAL
PARANÁ

- O atributo **showAsAction** não existe nas versões anteriores do Android
- A referência deve ser feita usando outro **namespace**

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

    <item
        ...
        myapp:showAsAction="ifRoom"
        ... />
```

O namespace android não pode ser utilizado

Menus

INSTITUTO FEDERAL
PARANÁ

- Menus são elementos gráficos que permitem que os usuários executem ações através de cliques
- O Android suporta diversos tipos de menus
  - **Options menu / Action bar**
  - **Popup menus**
  - **Context menu**

Options Menu / Action bar

INSTITUTO FEDERAL
PARANÁ

- A **action bar** pode ter um menu de opções
- Usar a **action bar** é a forma recomendada
- Em versões anteriores ao Android 3.0, é possível usar o **options menu**
  - Acessível através do botão "**Menu**" do dispositivo
- Em termos de programação, ambas as técnicas são iguais

Alteração do Menu

INSTITUTO FEDERAL
PARANÁ

- O método **`onCreateOptionsMenu()`** é chamado apenas uma vez
  - Isto faz com que o menu seja estático
- Para atualizar o menu de forma dinâmica, de acordo com outros parâmetros, é preciso implementar **`onPrepareOptionsMenu()`**

```
public boolean onPrepareOptionsMenu(Menu menu) {
    //...
    return super.onPrepareOptionsMenu(menu);
}
```

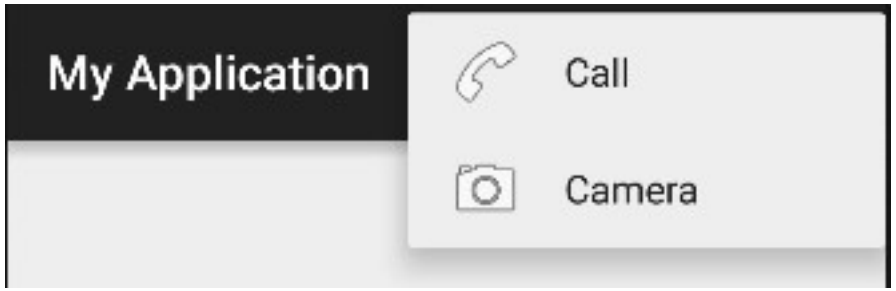
Alteração do Menu

INSTITUTO FEDERAL
PARANÁ

- Se a versão do Android for anterior à 3.0 (API Level 11), **`onPrepareOptionsMenu()`** é chamado toda vez que o **options menu** é aberto
- Se a versão do Android for 3.0 ou superior, é preciso chamar **`invalidateOptionsMenu()`** para que **`onPrepareOptionsMenu()`** seja invocado

Submenus

- O **options menu** e **action bar** podem contar com submenus



- A definição da hierarquia dos menus e submenus é feita no arquivo de layout

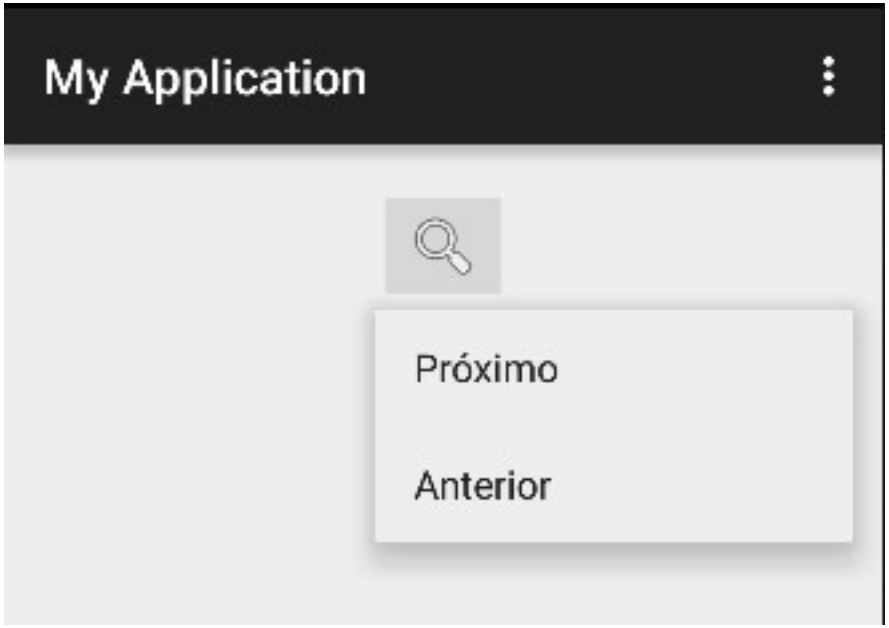
Submenus

INSTITUTO FEDERAL
PARANÁ

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/action_more"
        android:icon="@android:drawable/ic_menu_add"
        app:showAsAction="ifRoom"
        android:title="Actions">
        <menu>
            <item
                android:id="@+id/action_call"
                android:icon="@android:drawable/ic_menu_call"
                android:title="@string/call" />
            <item
                android:id="@+id/action_camera"
                android:icon="@android:drawable/ic_menu_camera"
                android:title="@string/camera" />
        </menu>
    </item>
</menu>
```

Popup Menus

- Também é um menu associado a uma **view**
- O **popup menu** deve ser usado para oferecer opções que não interferem na **view**
  - **Context menus** podem interferir na **view**



Popup Menus

INSTITUTO FEDERAL
PARANÁ

- Criar uma instância de **`PopupMenu`**

```
PopupMenu popup = new PopupMenu(this, view);
MenuInflater inflater = popup.getMenuInflater();
inflater.inflate(R.menu.menu_activity, popup.getMenu());
popup.show();
```

```
PopupMenu popup = new PopupMenu(this, view);
popup.inflate(R.menu.menu_activity);
popup.show();
```

Context Menus

INSTITUTO FEDERAL
PARANÁ

- São menus de contexto
  - Associados a uma ou mais **views** da tela
- São normalmente usados em listas de elementos
- Podem ser exibidos de duas formas
  - **Floating context menu**
  - **Contextual action mode**

Contextual Action Mode

INSTITUTO FEDERAL
PARANÁ

- **Floating context menus** eram usados até o android 2.3.3 (API Level 10). Do Android 3.0 em diante, é recomendado usar o **contextual action mode**
- Continua sendo um **menu de contexto** associado a uma ou mais **views**
- O menu aparece no topo da tela, no local onde fica a **action bar**

Contextual Action Mode

INSTITUTO FEDERAL
PARANÁ

- Pode ser de dois tipos
  - Associado a uma **view** qualquer
  - Associado a um **grupo de views** (Ex: ListView)

Action Mode em uma View

INSTITUTO FEDERAL
PARANÁ

- Primeiramente é preciso implementar a interface **`ActionMode.Callback`**

```
public class MainActivity extends Activity implements ActionMode.Callback
{

    public boolean onCreateActionMode(ActionMode mode, Menu menu) {
    }

    public boolean onPrepareActionMode(ActionMode mode, Menu menu) {
    }

    public boolean onActionItemClicked(ActionMode mode, MenuItem item) {
    }

    public void onDestroyActionMode(ActionMode mode) {
    }
}
```

Action Mode em uma View

INSTITUTO FEDERAL
PARANÁ

- Chamar o método **`startActionMode()`** quando for o momento de exibir a barra de menu de contexto

```
public void onCreate(Bundle savedInstanceState) {
    View view = findViewById(R.id.view);
    view.setOnLongClickListener(new  View.OnLongClickListener() {
        public boolean onLongClick(View v) {
            startActionMode(MainActivity.this);
            return true;
        }
    });
}
```

Action Mode em um Grupo de Views

INSTITUTO FEDERAL
PARANÁ

- É preciso implementar a interface **`AbsListView.MultiChoiceModeListener`**

```
public class MainActivity extends Activity implements
    AbsListView.MultiChoiceModeListener {

    public boolean onCreateActionMode(ActionMode mode, Menu menu) { }

    public boolean onPrepareActionMode(ActionMode mode, Menu menu) { }

    public boolean onActionItemClicked(ActionMode mode, MenuItem item){ }

    public void onItemCheckedStateChanged(ActionMode mode, int position,
        long id, boolean checked) { }

    public void onDestroyActionMode(ActionMode mode) { }
}
```

listView.setChoiceMode(AbsListView.CHOICE_MODE_MULTIPLE_MODAL)