Universidade Federal de Ouro Preto
Departamento de Computação e Sistemas - DECSI

Computação Móvel

ActionBar e Temas (Ref. Cap. 5)

Vicente Amorim
vicente.amorim.ufop@gmail.com
www.decom.ufop.br/vicente

Sumário

* Introdução

* *ActionBar*

* Temas

Introdução

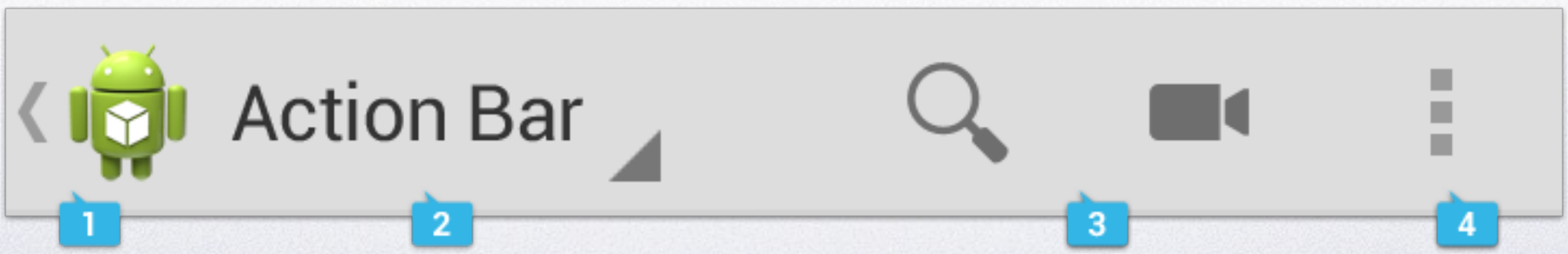
Introdução

- Visão geral

  ✓ *ActionBar* mostra de forma organizada ao usuário as possíveis ações a serem executadas a cada *Activity* da aplicação.

  ✓ Usuários de Android já se acostumaram com a mesma.

✓ Pode ser customizada de acordo com a *Activity* / App.

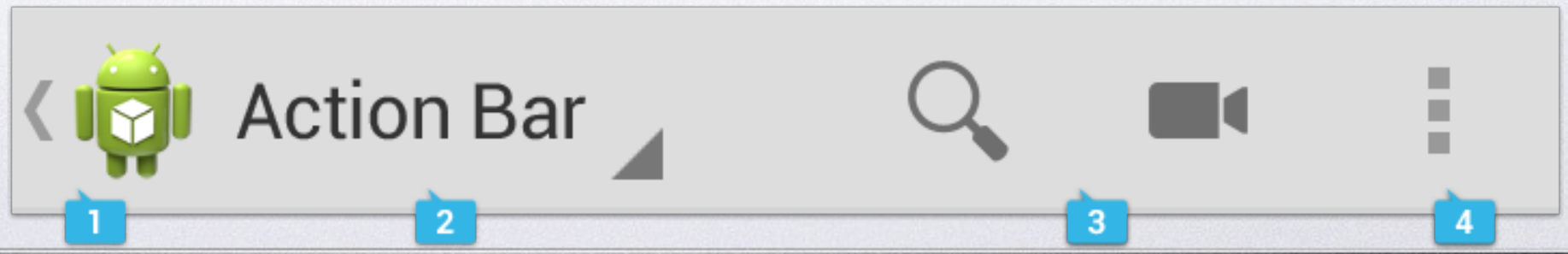


Introdução

- Visão Geral:

✓ 1 - Ícone de aplicação: Por padrão possui o ícone do projeto. Também pode exibir o "*up navigation*" (seta voltar).

. *Design:* Nas versões iniciais do Android o ícone reforçava o conceito de representação de alguma "ação importante". Do *material design* (Android 5.0 em diante) em diante o ícone foi suprimido dando-se um enfoque maior nas cores dos aplicativos. Desde então é uma boa prática utilizar cores para identificar a marca do cliente/app.



Introdução

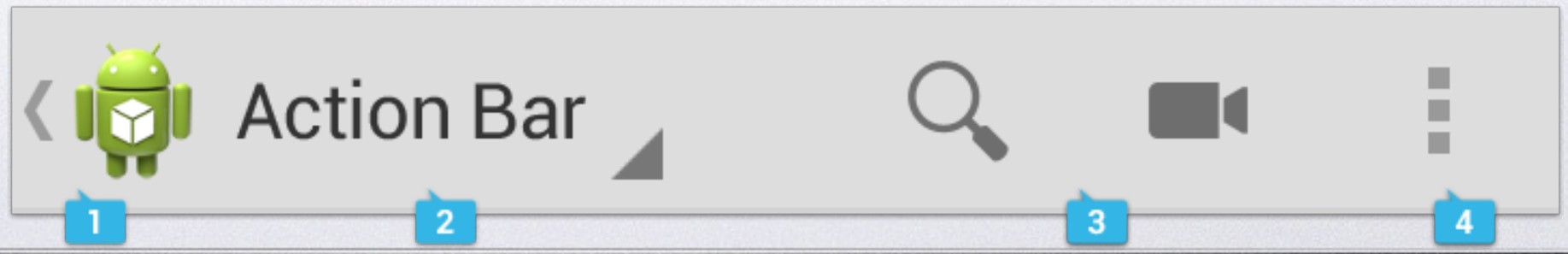
- Visão Geral:

  ✓ 2 - *View control*

  . Título do aplicativo ou da tela. Pode-se utilizar também algum controle de navegação ou tabs.

  ✓ 3 - Botões de ação

  . Botões que representam ações mais comuns/importantes do aplicativo. Caso a quantidade de botões seja maior que o espaço, esses serão inseridos automaticamente no *action overflow*.

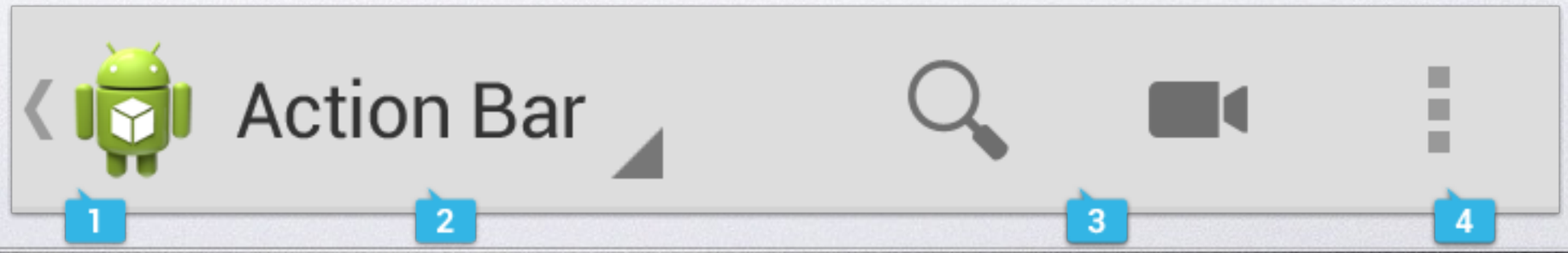


Introdução

- Visão Geral:

  ✓ 4 - *Action overflow*

  . Menu flutuante que mostra as ações não tão comuns/importantes do aplicativo.



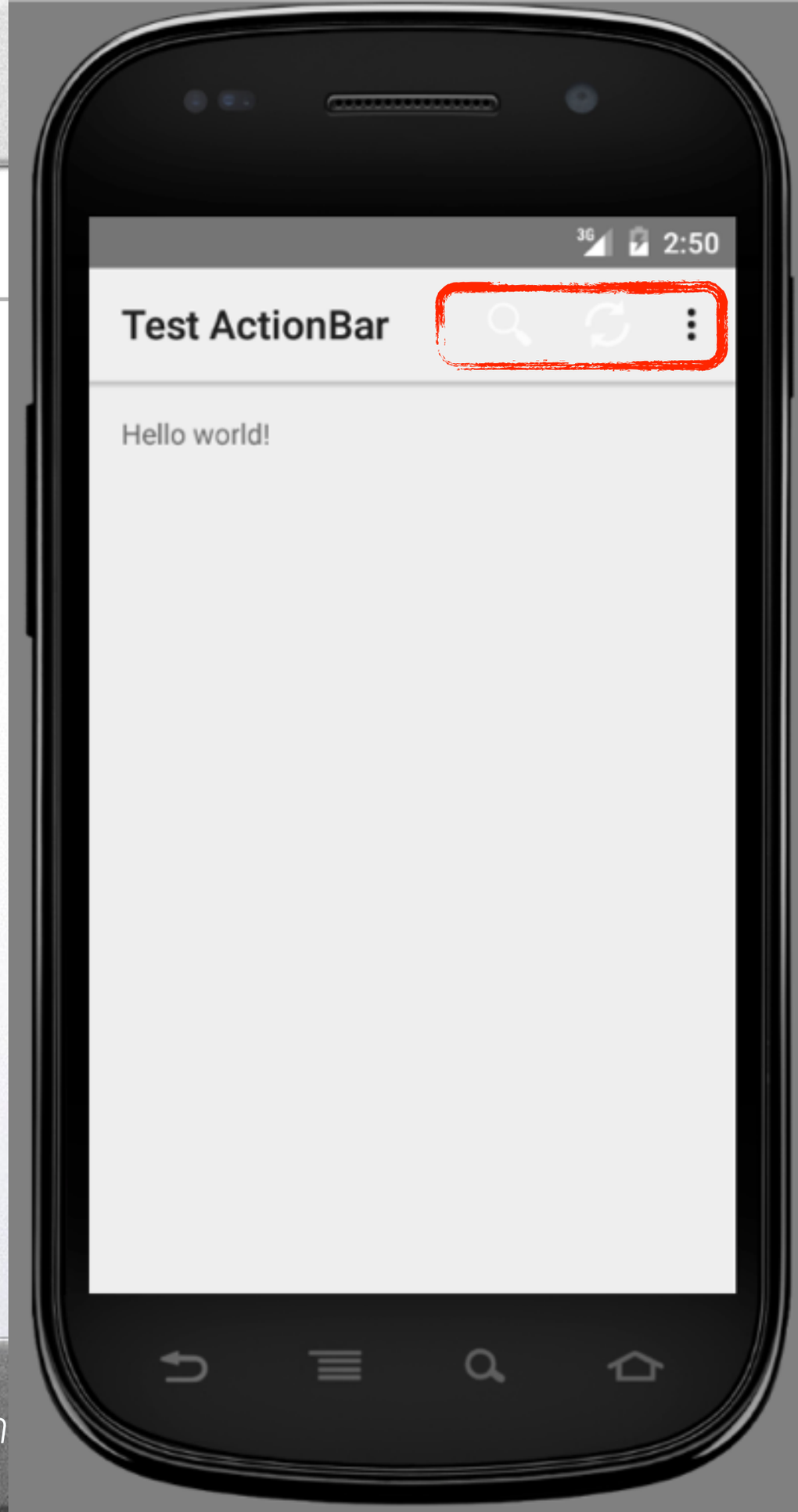
ActionBar

ActionBar

- Exemplo: AppNum13

  ✓ Executando o exemplo é possível ver que o mesmo possui uma *ActionBar*.

  ✓ Existem dois botões posicionados diretamente na *ActionBar* e um terceiro no *action overflow*.

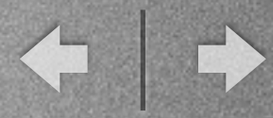



ActionBar

- Exemplo: AppNum13

✓ Os arquivos de menu são definidos no arquivo XML */res/menu/menu_main.xml*.

```
<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_search"
    android:icon="@drawable/ic_action_search"
    android:title="@string/action_search"
    android:showAsAction="always" />

    <item
    android:id="@+id/action_refresh"
    android:icon="@drawable/ic_action_refresh"
    android:title="@string/action_refresh"
    android:showAsAction="always" />
    <item
    android:id="@+id/action_settings"
    android:title="@string/action_settings"
    android:showAsAction="never" />
</menu>
```

ActionBar

- Exemplo: AppNum13

✓ Para que o menu se torne funcional, a implementação do tratamento deve existir dentro da *Activity*.

MainActivity.java

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

    ActionBar actionBar = getActionBar();
    actionBar.setTitle("Test ActionBar");

}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Infla o menu com os botões da action bar
    getMenuInflater().inflate(R.menu.menu_main, menu);

    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.action_search) {
        toast("Clicou no Search!");
        return true;
    } else if (id == R.id.action_refresh) {
        toast("Clicou no Refresh!");
        return true;
    } else if (id == R.id.action_settings) {
        toast("Clicou no Settings!");
        return true;
    }
    return super.onOptionsItemSelected(item);
}
```

- Visualização dos botões na *ActionBar*

✓ Constantes permitidas no campo android:showAsAction do *layout* do menu.

. *always*: Botão deve sempre ficar visível como *action button*. Ações mais comuns do app são definidas dessa forma.

. *ifRoom*: Mostra o botão na *ActionBar* se existir espaço. Senão, move-o automaticamente para *action overflow*.

. *withText*: Mostra o título do botão ao lado do ícone caso haja espaço suficiente.

. *never*: Nunca mostra na barra e sempre no *action overflow*.

. *collapseActionView*: Indica que a *view* relacionada é grande, logo inicialmente deve mostrar o botão de forma contraída (*ex.: search)*.

- Ícones para os botões da *ActionBar*

✓ Ícones e cores devem combinar de forma a evitar possíveis problemas de usabilidade (como na AppNum13).

✓ Cada item do menu referencia um ícone diferente através do atributo *android:icon*.

✓ Um pacote comum de ícones mais utilizados pode ser baixado diretamente do Google.

http://developer.android.com/design/patterns/actionbar.html

. Item: "*Download the Action Bar Icon Pack*".

Classe *android.app.ActionBar*

Classe *android.app.ActionBar*

- Muitas das opções relacionadas a *ActionBar* podem ser configuradas via *AndroidManifest.xml*.

- Entretanto, a API da própria classe fornece funções para a configuração da *ActionBar* em tempo de execução.

✓ *setCustomView(int / View)*: Permite adicionar uma *view* customizada na *ActionBar* (por exemplo um botão de busca).

✓ *setTitle(string)*: Altera o título de *ActionBar*.

✓ *setIcon(Drawable)*: Altera o ícone relacionado da *ActionBar*.

✓ *setDisplayShowTitleEnabled(bool)*: Configura se o título será ou não exibido.

Classe *android.app.ActionBar*

- Entretanto, a API da própria classe fornece funções para a configuração da *ActionBar* em tempo de execução. (cont.)

  ✓ *setDisplayShowHomeEnabled(boolean)*: Configura se é necessário exibir ou não o ícone na *ActionBar*.

  ✓ *setDisplayHomeAsUpEnabled(boolean)*: Exibe o esconde a seta "*back*" conhecida como *up navigation*.

Classe *android.app.ActionBar*

- *SearchView* - AppNum14

  ✓ A função *setCustomView(…)* permite a adição de uma *view* customizada na *ActionBar*.

/res/menu/menu_main.xml

```
<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_search"
    android:icon="@drawable/ic_action_search"
    android:title="@string/action_search"
    android:showAsAction="always"
    android:actionViewClass="android.widget.SearchView"/>
...
</menu>
```

Classe *android.app.ActionBar*

- *SearchView* - AppNum14

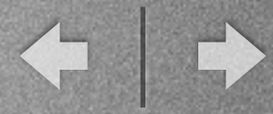
  ✓ A função *setCustomView(…)* permite a adição de uma *view* customizada na *ActionBar*. *(cont.)*

MainActivity.java

```
public class MainActivity extends Activity {

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

        ActionBar actionBar = getActionBar();
        actionBar.setTitle("CEA 436");

    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Infla o menu com os botões da action bar
        getMenuInflater().inflate(R.menu.menu_main, menu);

        MenuItem item = menu.findItem(R.id.action_search);
        SearchView searchView = (SearchView) item.getActionView();
        searchView.setOnQueryTextListener(onSearch());

        return true;
    }

    private SearchView.OnQueryTextListener onSearch() {
        return new SearchView.OnQueryTextListener(){
            @Override
            public boolean onQueryTextSubmit(String query) {
                // Usuário fez a busca
                toast("Buscar o texto: " + query);
                return false;
            }
            @Override
            public boolean onQueryTextChange(String newText) {
                // Mudou o texto digitado
                return false;
            }
        };
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.action_search) {
            toast("Clicou no Search!");
            return true;
        } else if (id == R.id.action_refresh) {
            toast("Clicou no Refresh!");
            return true;
        } else if (id == R.id.action_settings) {
            toast("Clicou no Settings!");
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
```

Classe *android.app.ActionBar*

- *SearchView* - AppNur

Classe *android.app.ActionBar*

- *ActionProvider*

✓ Também inclui na *ActionBar* um botão customizado.

✓ *ActionProvider* é configurado na *ActionBar* através da inserção da *tag* *android:actionProviderClass*.

✓ Na *tag* deve-se informar alguma subclasse de *android.view.ActionProvider*.

✓ Faz com que seja possível incorporar várias novas funcionalidades à *ActionBar*.

Classe *android.app.ActionBar*

- *ActionProvider* - AppNum15

✓ Suporte da ação de "compartilhar" (*share*) via *ActionBar*.

/res/menu/menu_main.xml

```
...
<item
android:id="@+id/action_share"
android:actionProviderClass="android.widget.ShareActionProvider"
android:icon="@drawable/ic_action_share"
android:showAsAction="always"
android:title="@string/action_share" />
...
```

- *ActionProvider (cont.)*

✓ No MainActivity.java, após configurar o item de menu da *ActionBar* com o *ShareActionProvider*, basta customizar a *Intent* que será utilizada para disparar a mensagem para SO afim de compartilhar o conteúdo.

✓ Uma *Intent* é uma espécie de mensagem enviada ao SO Android para os mais variados fins.

✓ Em nosso caso, a *Intent* é enviada e o SO vai perguntar para todas as aplicações instaladas quem é capaz de tratar o "compartilhamento daquele tipo de conteúdo".

Classe *android.app.ActionBar*

- *ActionProvider* - AppNum15

  ✓ Suporte da ação de "compartilhar" (*share*) via *ActionBar*. (cont.)

MainActivity.java

```
...
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Infla o menu com os botões da action bar
    getMenuInflater().inflate(R.menu.menu_main, menu);

    // SearchView
    MenuItem item = menu.findItem(R.id.action_search);
    SearchView searchView = (SearchView) item.getActionView();
    searchView.setOnQueryTextListener(onSearch());

    // ShareActionProvider
    MenuItem shareItem = menu.findItem(R.id.action_share);
    ShareActionProvider share = (ShareActionProvider) shareItem.getActionProvider();
    share.setShareIntent(getDefaultIntent());

    return true;
}

// Intent que define o conteúdo que será compartilhado
private Intent getDefaultIntent() {
    Intent intent = new Intent(Intent.ACTION_SEND);
    intent.setType("text/*");
    intent.putExtra(Intent.EXTRA_TEXT, "Texto para compartilhar");
    return intent;
}
...
```

Classe *android.app.ActionBar*

- *ActionProvider* - AppNum

  ✓ Suporte da ação de “compartilhar” (*share*) via *ActionBar*. (cont.)

Classe *android.app.ActionBar*

- *SplitActionBar*

  ✓ Dependendo da quantidade de ações principais do sistema, é necessário redistribuir melhor seu *layout*.

  ✓ Uma opção é colocar botões na parte inferior da Activity utilizando o *SplitActionBar*.

  ✓ Usabilidade deve ser considerada no momento da escolha de quais botões irão para a parte de baixo.

  ✓ Importante: Funcionalidade **descontinuada** para o Android 5. Não suportada pelo *Material Design*.

Classe *android.app.ActionBar*

- *SplitActionBar - AppNum16*

  ✓ Alterações específicas devem ser feitas na referência à atividade no *AndroidManifest.xml*.

AndroidManifest.xml

```
...
<activity
    android:name=".MainActivity"
    android:label="@string/app_name"
    android:uiOptions="splitActionBarWhenNarrow">
    <meta-data android:name="android.support.UI_OPTIONS" android:value="splitActionBarWhenNarrow"/>
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
...
```

Classe *android.app.ActionBar*

- *UpNavigation*

  ✓ Identificado pelo ícone da "seta para esquerda" e é utilizado para subir na hierarquia de telas.

  ✓ O comportamento padrão do *up navigation* é voltar para a tela anterior.

  ✓ É possível configurar o *AndroidManifest.xml* para que a "volta" para a tela anterior seja padrão.

  ✓ Atributo *android:parentActivityName* permite configurar a *Activity* mãe na hierarquia das telas.

  ✓ *Tag* <meta…> também precisa ser configurada para suporte a versões API Level 7 e superiores.


Classe *android.app.ActionBar*

- *UpNavigation*

AndroidManifest.xml

```
...
<activity android:name=".BemVindoActivity" android:label="@string/
title_bem_vindo_activity"
android:parentActivityName="com.cea436.appnum8.MainActivity" >
    <!-- Parent activity meta-data to support API level 7+ -->
    <meta-data
    android:name="android.support.PARENT_ACTIVITY"
    android:value="com.cea436.appnum8.MainActivity" />
</activity>
...
```

Classe *android.app.ActionBar*

- *ActionBar com Tabs*

✓ O uso de abas permite que aplicativos exibam várias telas com informações relevantes aos usuários.

✓ Devem ser utilizadas para o caso de existirem poucas seções.

✓ O objetivo final é prover ao usuário acesso rápido e prático a todo o aplicativo.

✓ É válido analisar o design dos principais aplicativos da Google Play como forma de aprendizado.

Classe *android.app.ActionBar*

- *ActionBar com Tabs*

✓ A criação das *tabs* (abas) se dá através do uso do método *setNavigationMode(ActionBar.NAVIGATION_MODE_TABS)*.

✓ Cada invocação ao método *addTab(x)* cria uma nova aba.

✓ Ao selecionar uma *tab*, os *callbacks* da *ActionBar.TabListener* são invocados para permitir que o aplicativo realize a operação esperada.

✓ Importante: API já *deprecated* no Android 5 devido à introdução do *TabLayout* e *ToolBar*.

Classe *android.app.ActionBar*

- *ActionBar com Tabs - AppNum17*

MainActivity.java

```
...
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ActionBar actionBar = getActionBar();
        actionBar.setTitle("Hello ActionBar");

        actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);
        // Cria as tabs (Passa como parâmetro o índice de cada tab: 1,2,3)
        actionBar.addTab(actionBar.newTab().setText("Tab 1").setTabListener(new MyTabListener(this,1)));
        actionBar.addTab(actionBar.newTab().setText("Tab 2").setTabListener(new MyTabListener(this,2)));
        actionBar.addTab(actionBar.newTab().setText("Tab 3").setTabListener(new MyTabListener(this,3)));
    }
...
```


Classe *android.app.ActionBar*

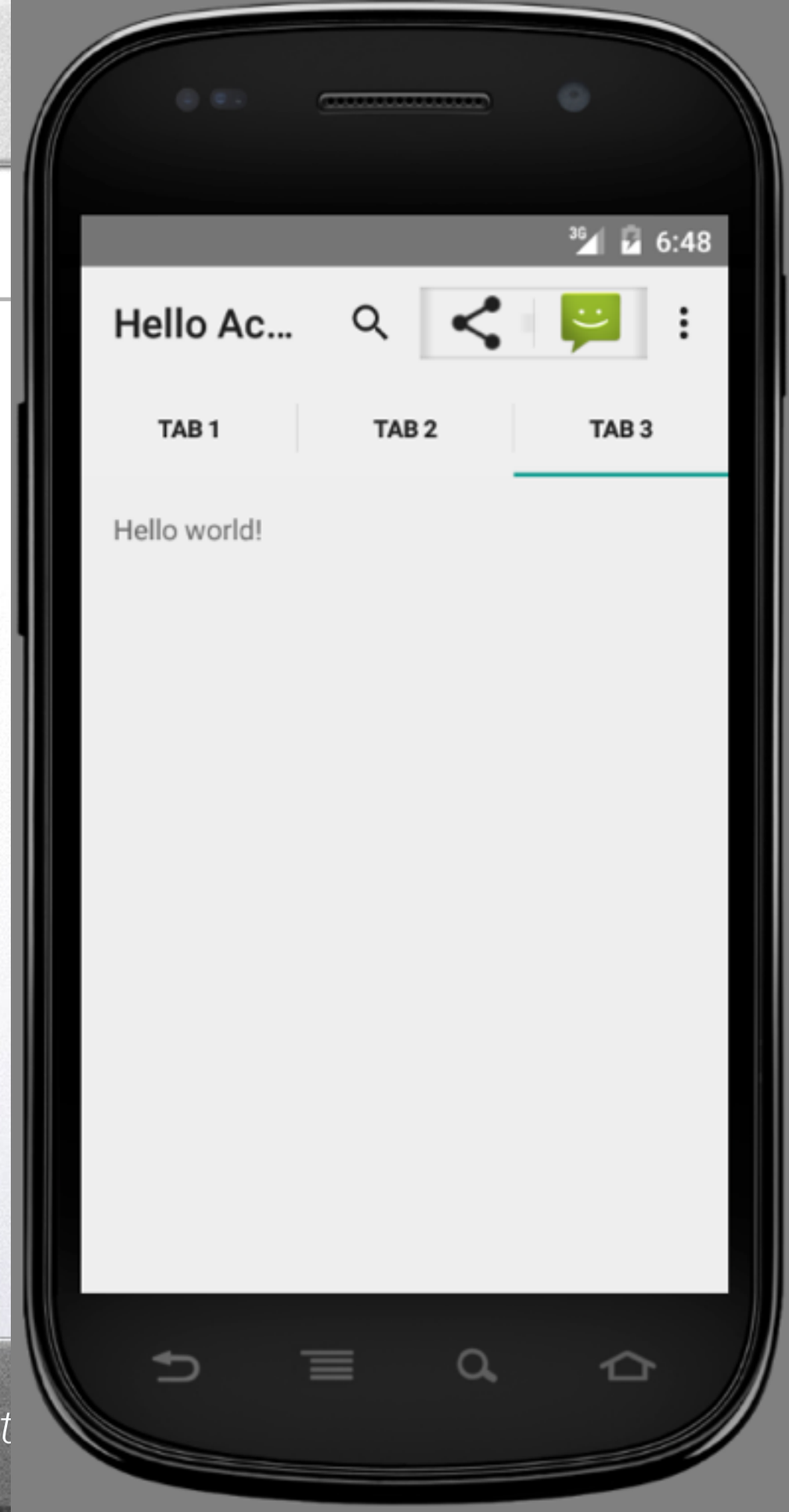
- *ActionBar com Tabs - AppNum17*

MyTabListener.java

```
...
public class MyTabListener implements ActionBar.TabListener {
    private Context context;
    private int tabIdx;

    public MyTabListener(Context context, int tabIdx) {
        this.context = context;
        this.tabIdx = tabIdx;
    }

    @Override
    public void onTabSelected(ActionBar.Tab tab, FragmentTransaction ft) {
        // Chamado ao selecionar uma tab
        Toast.makeText(context, "Selecionou a tab: " + tabIdx, Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onTabUnselected(ActionBar.Tab tab, FragmentTransaction ft) {
        // Chamado quando a tab perde o foco (se outra tab é selecionada)
    }

    @Override
    public void onTabReselected(ActionBar.Tab tab, FragmentTransaction ft) {
        // Chamado quando uma tab é selecionada novamente.
    }
}
```

Classe *android.app.ActionBar*

- *ActionBar com Tabs - AppNum17*



Classe *android.app.ActionBar*

- Exercícios:

9) Implementar a busca de texto (*search*) relativa ao botão adicionado na *ActionBar*. (AppNum14)

10) Pesquisar e reescrever a AppNum17 utilizando a API *ToolBar*.

Temas

Temas

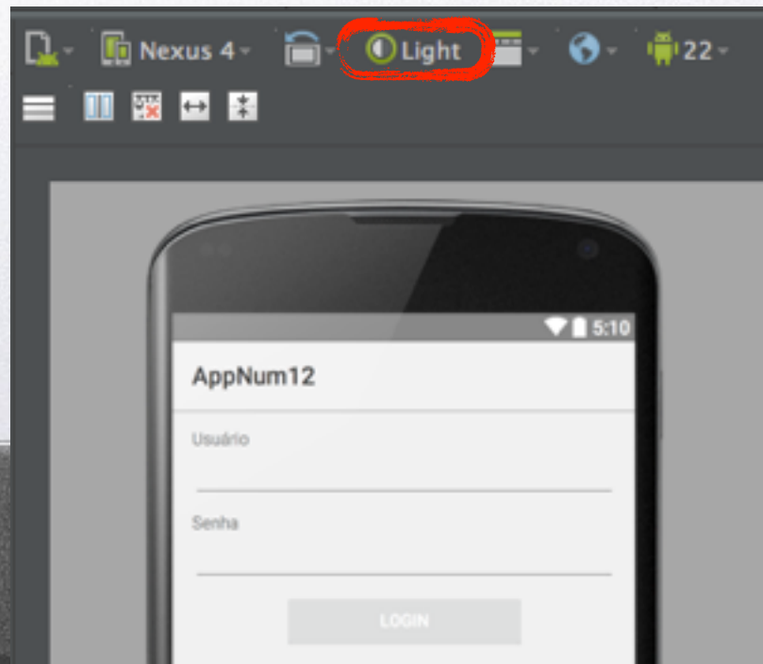
- Introdução:

✓ Até o Android 2.x existiam apenas os temas Black (*Theme.Black*) e Light (*Theme.Light*).

✓ No Android 3.0 foi introduzido o tema *Holographic* (*Theme.Holo* e *Theme.Holo.Light*).

✓ O Android 5 trouxe o tema Material (*Theme.Material* e *Theme.Material.Light*).

✓ Temas podem ser configurados diretamente no editor de *layout*.



Temas

- Introdução:

✓ Android permite a customização dos temas através da edição do arquivo */res/values/styles.xml/styles.xml*.

✓ O tema da aplicação é definido no arquivo *AndroidManifest.xml*.

```
...
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
...
```