

Action Bar

Barra d'accions

Programació Multimèdia i Dispositius Mòbils

2n. DAM



Action Bar

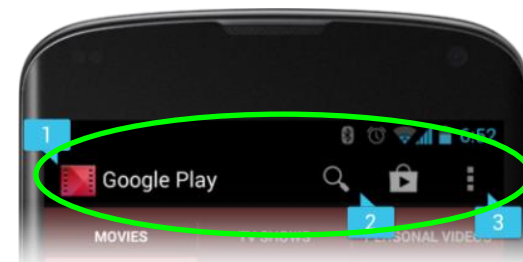
- Què és?

<http://developer.android.com/guide/topics/ui/actionbar.html>

L'Action Bar és una característica de les finestres que proporciona a l'usuari accions i maneres de navegar.

- **Aspectes clau que ens proporciona l'Action Bar:**

- 1.-Ens proporciona un espai per a identificar la nostra app, i indicar a l'usuari en quin lloc(apartat) de l'app es troba.
- 2.-ens proporciona una zona on col·locar eines que s'utilitzen sovint, com per exemple cercar, així com ítems de menús (vegeu l'apartat de menús).
- 3.-ens proporciona una navegació consistent, i ens permet canviar entre distints apartats de la nostra app.



- L'api de l'Action Bar es va introduir a partir de la versió 3.0 d'Android. Per a versions anteriors haureu d'utilitzar la llibreria de suport ([Support Library](#)) d'Android.

Descarregueu les icones d'Android per a l'Action Bar

https://developer.android.com/downloads/design/Android_Design_Icons_20131106.zip

Podeu trobat més info sobre l'iconografia en Android a l'url

<https://developer.android.com/design/style/iconography.html>



Action Bar

- Què podem fer amb l'Action Bar?
 - Mostrar i Ocultar-la
 - Mostrar un logo en lloc d'una icona
 - Separar-lo en dues parts
 - Navegar amb l'Icona de l'app
 - Afegir un element d'acció
 - Afegir un Proveïdor d'accions
 - Donar-li un estil
 - Superposar la barra d'accions
 - Afegir-li un Navigation Drawer.
 -

Action Bar. Mostrar i Ocultar-la



Com Afegir l'ActionBar:

- La barra d'accions s'inclou en totes les *activities* que utilitzen el tema Theme.Holo (o un dels seus descendents), que és el tema per defecte quan s'utilitza un targetSdkVersion o un minSdkVersion amb un valor d'11 o superior.
- En cas que volguerem utilitzar un tema que no mostrara cap l'ActionBar, podriem establir el tema de l'Activity com Theme.Holo.NoActionBar.

Com Ocultar l'ActionBar

Per a mostrar/ocultar la barra d'accions caldrà seguir els següents passos:

- Obtindrem una referència a l'ActionBar de l'activity i després cridarem als mètodes hide() o show() per a ocultar-la o mostrar-la (respectivament)

```
ActionBar ab = getActionBar(); //Obtenim una instància
                                // de l'ActionBar d'aquest Activity
ab.hide(); //Ens oculta l'ActionBar
ab.show(); //Ens mostra l'ActionBar
```

Atenció: Cada vegada que ocultem i/o mostrem l'ActionBar, Android reajusta els layouts per a ocupar tot l'espai disponible.



Action Bar. Mostrar un logo en lloc d'una icona

- Per defecte, l'ActionBar utilitza la icona de l'app. Si en compte de la icona voleu posar un logo de manera que s'identifique millor la vostra app, ho podeu fer mitjançant l'atribut logo del tag `<application>` o `<activity>` de l'android Manifest.

`android:logo="@drawable/logotip"`

Un logotip per a l'aplicació en el seu conjunt, i el logotip predeterminat per a les activitats.

Aquest atribut s'ha d'establir com una referència a un recurs drawable que conté la imatge (per exemple "`@drawable/logotip`").

No hi ha logotip per defecte.

Action Bar. Separar-lo en dues parts



La Barra d'accions ens ofereix la possibilitat de tindre un altra barra en la part inferior de la pantalla per mostrar tots els elements d'acció quan l'activity s'està executant en una pantalla estreta (com un telèfon amb orientació vertical).

Aquesta separació ens assegura una quantitat raonable d'espai per mostrar tots els seus elements d'acció en una pantalla estreta, deixant l'espai superior als elements de navegació i el títol.

Per activar l'ActionBar separada



Afegir `uiOptions = "splitActionBarWhenNarrow"` a cada element `<activity>` o a l'element `<application>`. Aquest atribut només s'entén per a l'API nivell 14 i superiors(i és ignorat per les versions anteriors).

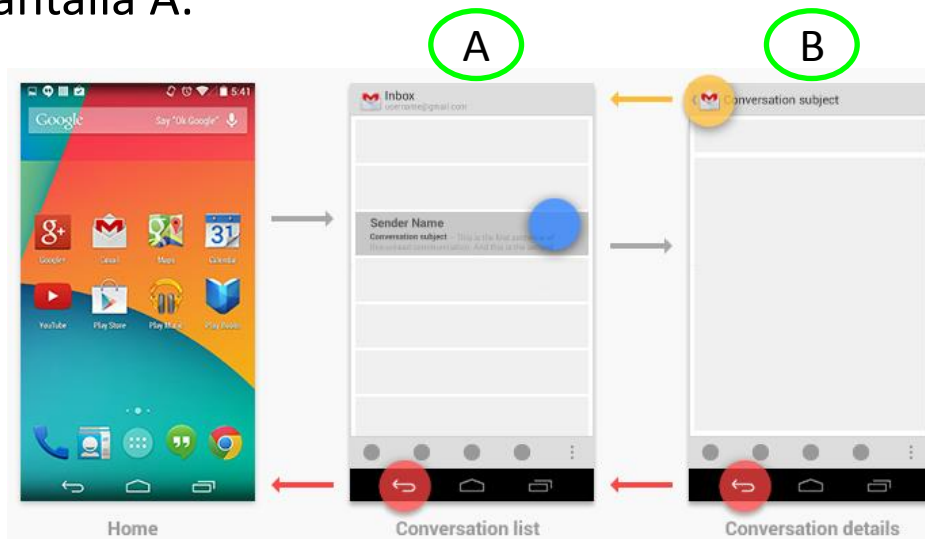
- Per donar suport a les versions anteriors, caldrà afegir un element `<meta-data>` com a fill de cada `<activitat>` que declare el mateix valor per a `"android.support.UI_OPTIONS"`.

```
<manifest ...>
  <activity uiOptions="splitActionBarWhenNarrow" ... >
    <meta-data android:name="android.support.UI_OPTIONS"
      android:value="splitActionBarWhenNarrow" />
  </activity>
</manifest>
```

Action Bar. Navegar amb l'Icona de l'app



- <http://developer.android.com/training/implementing-navigation/ancestral.html>
- Habilitar la icona de l'aplicació com un botó Amunt permet a l'usuari navegar a la seva aplicació sobre la base de les relacions jeràrquiques entre pantalles. Per exemple, si la pantalla A mostra una llista d'elements, i seleccionar un element porta a la pantalla B, llavors la pantalla B ha d'incloure el botó Amunt, que torna a la pantalla A.



- Podeu veure més qüestions sobre el disseny de la navegació en <http://developer.android.com/design/patterns/navigation.html>

Action Bar. Navegar amb l'Icona de l'app



- Per a activar la icona de l'app com un botó *Amunt* cal cridar al mètode [setDisplayHomeAsUpEnabled\(\)](#) i especificar-li les accions a desencadenar.
- ```
ActionBar actionBar = getSupportActionBar();
actionBar.setDisplayHomeAsUpEnabled(true);
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_details);
 ...
 ActionBar actionBar = getActionBar();
 actionBar.setDisplayHomeAsUpEnabled(true); // Activem la icona com a botó Amunt
 ...
}
```



# Action Bar. Navegar amb l'Icona de l'app



Una vegada ja li hem indicat a la icona que actue com a botó de navegació, cal definir-li les accions que ha de desenvolupar (tenim 2 opcions):

## 1. Especificar l'activity '*pare*' al fitxer Manifest.

A partir d'Android 4.1 (nivell d'API 16), es pot declarar el pare lògic de cada activity especificant l'atribut **android: parentActivityName** a l'element `<activity>`.

Si l'aplicació és compatible amb Android 4.0 o inferiors, inclogueu la biblioteca de suport al vostre projecte i afegiu un element `<meta-data>` dins de l'element `< activity >`. A continuació, especifiqueu l'activitat del pare com el valor per `android.support.PARENT_ACTIVITY`, que coincideix amb l'atribut `android: parentActivityName`.

```
<application ... >
...
<!-- L'activity principal/inicial (aquest no té activity 'pare') -->
<activity
 android:name="com.example.myfirstapp.MainActivity" ...>
...
</activity>
<!-- Un activity fill de l'activity principal -->
<activity
 android:name="com.example.myfirstapp.DisplayMessageActivity"
 android:label="@string/title_activity_display_message"
 android:parentActivityName="com.example.myfirstapp.MainActivity" >
 <!-- meta-data de l'Activity 'pare' per a versions 4.0 i inferiors -->
 <meta-data
 android:name="android.support.PARENT_ACTIVITY"
 android:value="com.example.myfirstapp.MainActivity" />
</activity>
</application>
```

# Action Bar. Navegar amb l'Icona de l'app



Una vegada modificat el `AndroidManifest.xml`, per a poder navegar cap amunt quan l'usuari prem la icona de l'aplicació, s'ha d'utilitzar el mètode estàtic de la classe `NavUtils`, **`navigateUpFromSameTask()`**. Quan s'executa aquest mètode, s'acaba l'activity actual i s'inicia l'activity pare. Si l'activitat és pare objecte a la pila l'esquena de la tasca, que s'avança

**2.- Sobreescriure els mètodes `getSupportParentActivityIntent()` i `onCreateSupportNavigateUpTaskStack()` a l'activity.**

Aquesta opció és adequada quan l'activity pare pot ser diferent depenent de com l'usuari haja arribat a la pantalla actual.

És a dir, si hi ha molts camins que l'usuari podria haver fet per arribar a la pantalla actual, el botó *Amunt* hauria de navegar cap enrere al llarg de la ruta d'accés que l'usuari va seguir per arribar-hi.

# Action Bar. Afegir un element d'acció



Un Element d'acció és una element que apareix en la Barra d'accions i que actua com si fos un botó. Aquest element ens proporciona un accés ràpid a les accions més habituals, sense haver de canviar d'activity o de fragment. (Vegeu la unitat dedicada als Menus).



```
<?xml version="1.0" encoding="utf-8"?>
<menu
 xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:yourapp="http://schemas.android.com/apk/res-auto" >
 <item android:id="@+id/action_search"
 android:title="@string/action_search"
 android:icon="@drawable/ic_action_search"
 yourapp:showAsAction="ifRoom|collapseActionView"
 yourapp:actionViewClass="android.support.v7.widget.SearchView" />
</menu>
```

# Action Bar. Afegir un element d'acció



Per a afegir-li Listeners a l'element d'acció, ho farem dins el mètode **onCreateOptionsMenu ()**. Hem d'instanciar l'element d'acció cridant al mètode estàtic **MenuItemCompat.getActionView()** i passar-li el MenuItem corresponent. Per exemple, l'element de cerca de l'exemple anterior s'adquireix d'aquesta manera:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
 getMenuInflater().inflate(R.menu.main_activity_actions, menu);
 MenuItem searchItem = menu.findItem(R.id.action_search);
 SearchView searchView = (SearchView) MenuItemCompat.getActionView(searchItem);
 // Configurem la cerca i els listener de l'element d'acció

 ...
 return super.onCreateOptionsMenu(menu);
}
```

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
 getMenuInflater().inflate(R.menu.main_activity_actions, menu);
 SearchView searchView = menu.findItem(R.id.action_search).getActionView();
 // Configurem la cerca i els listener de l'element d'acció

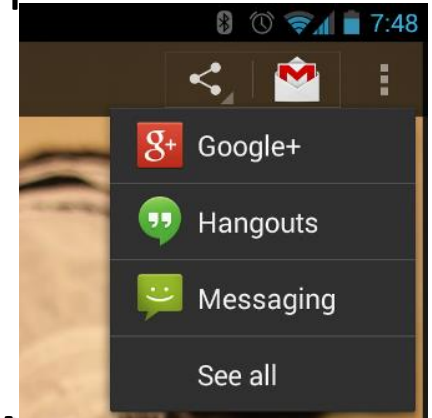
 ...
 return super.onCreateOptionsMenu(menu);
}
```

Per a  
versions de  
l'api 11 i  
superiors



# Action Bar. Afegir un Proveïdor d'accions

- Un proveïdor d'acció reemplaça un botó d'acció amb un disseny personalitzat i pot mostrar un submenú quan es pressiona.
- Només cal afegir un atribut **actionViewClass** al menú `<item>` amb un nom de classe complet per a un `ActionProvider`.
- Android ens proporciona alguns proveïdors d'acció pre-construïts com és el **ShareActionProvider**, cosa que facilita l'acció de "compartir" al mostrar una llista de possibles aplicacions per compartir directament dins la barra d'accions.



```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:yourapp="http://schemas.android.com/apk/res-auto" >
 <item android:id="@+id/action_share"
 android:title="@string/share"
 yourapp:showAsAction="ifRoom"
 yourapp:actionProviderClass="android.support.v7.widget.ShareActionProvider" />
 ...
</menu>
```



# Action Bar. Donar-li un estil

- Per a donar-li un estil més personalitzat, podem modificar el color de la barra d'acció, els colors del text, estils de botons, i més. Per a això, és necessari utilitzar marc estil i el tema d'Android per canviar l'estil de la barra acció utilitzant propietats d'estil especials.
- Crearem un nou fitxer XML amb el nou estil.
- Algunes característiques de l'element [actionBarStyle](#):
- [background](#) Defineix un recurs estirable per al fons de la barra.
- [backgroundSplit](#) Defineix un recurs drawable per a la barra d'acció de divisió
- [actionButtonStyle](#) Defineix un recurs d'estil dels botons s'acció.
- [actionOverflowButtonStyle](#) Defineix un recurs d'estil per als elements d'acció de desbordament.
- [displayOptions](#) Defineix una o més opcions de visualització de la barra d'accions, com ara la possibilitat d'utilitzar el logotip d'aplicació, mostrar el títol de l'activitat, o habilitar l'acció de dalt
- [divider](#) Defineix un recurs estirable per el divisor entre els punts d'acció.
- [titleTextStyle](#) Defineix un recurs d'estil pel títol de la barra d'acció.

<http://developer.android.com/guide/topics/ui/actionbar.html>





# ElMeuTemaActionBar

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
 <!-- the theme applied to the application or activity -->
 <style name="ElMeuTemaActionBar"
 parent="@style/Theme.AppCompat.Light">
 <item name="android:actionBarStyle">@style/MyActionBar</item>
 <item name="android:actionBarTabTextStyle">@style/TabTextStyle</item>
 <item name="android:actionMenuTextColor">@color/actionbar_text</item>

 <!-- Support library compatibility -->
 <item name="actionBarStyle">@style/MyActionBar</item>
 <item name="actionBarTabTextStyle">@style/TabTextStyle</item>
 <item name="actionMenuTextColor">@color/actionbar_text</item>
 </style>

 <!-- general styles for the action bar -->
 <style name="MyActionBar"
 parent="@style/Widget.AppCompat.ActionBar">
 <item name="android:titleTextStyle">@style/TitleTextStyle</item>
 <item name="android:background">@drawable/actionbar_background</item>
 <item name="android:backgroundStacked">@drawable/actionbar_background</item>
 <item name="android:backgroundSplit">@drawable/actionbar_background</item>

 <!-- Support library compatibility -->
 <item name="titleTextStyle">@style/TitleTextStyle</item>
 <item name="background">@drawable/actionbar_background</item>
 <item name="backgroundStacked">@drawable/actionbar_background</item>
 <item name="backgroundSplit">@drawable/actionbar_background</item>
 </style>
```

Aquest XML l'anonem:  
res/values/noutema.xml

## IMPORTANT:

Cal declarar un tema pare adequat,  
a partir del qual personalitzarem  
alguns atributs!!!

```
<!-- action bar title text -->
<style name="TitleTextStyle"
 parent="@style/TextAppearance.AppCompat.Widget.ActionBar.Title">
 <item name="android:textColor">@color/actionbar_text</item>
</style>

<!-- action bar tab text -->
<style name="TabTextStyle"
 parent="@style/Widget.AppCompat.ActionBar.TabText">
 <item name="android:textColor">@color/actionbar_text</item>
</style>
</resources>
```

# Action Bar. Donar-li un estil



- Afortunadament, podem trobar una eina que ens facilita la creació dels estils personalitzats:
- <http://romannurik.github.io/AndroidAssetStudio/>
- Una vegada generat el fitxer d'estil, al Manifest, haurem de referenciar-lo, bé a tota l'aplicació

```
<application android:theme="@style/ElMeuTemaActionBar" ... />
```

**Bé activity per activity:**

```
<activity android:theme="@style/ElMeuTemaActionBar" ... />
```





# Action Bar. Superposar L'ActionBar

- Per evitar el canvi de mida del seu disseny quan la barra d'acció s'oculta i es mostra, podem activar el mode de superposició de la barra d'acció.
- En aquest mode, l'activity utilitza tot l'espai disponible, com si la barra d'acció no hi fora i el sistema posa la barra d'acció 'davant'.
- Quan la barra d'acció s'oculta o apareix, el sistema no necessita canviar la mida del seu layout, i la transició és perfecta.



# Action Bar. Superposar L'ActionBar

- **Activar el mode Superposició.**

- Version 3.0 o superiors (api 11 i superiors): Cal crear-nos un tema personalitzat que herete un estil d'action bar, i afegir-li l'atribut: `android:windowActionBarOverlay` a "true"

```
<resources>
 <!-- the theme applied to the application or activity -->
 <style name="ElMeuTemaActionBar"
 parent="@style/Theme.AppCompat.Light">
 <item name="android:windowActionBarOverlay">true</item>
 <item name="android:actionBarStyle">@style/MyActionBar</item>
 ...
 <!-- Support library compatibility -->
 <item name="windowActionBarOverlay">true</item>
 <item name="actionBarStyle">@style/MyActionBar</item>
 <item name="actionBarTabTextStyle">@style/TabTextStyle</item>
 <item name="actionMenuTextColor">@color/actionbar_text</item>
 </style>
```

} Només en versions inferiors a la 3.0

# Action Bar. Navigation Drawer



- <http://creandoandroid.es/implementar-navegacion-drawer-menu-lateral/>