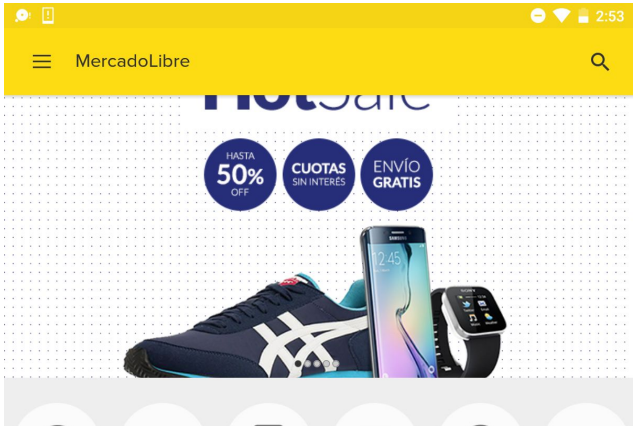
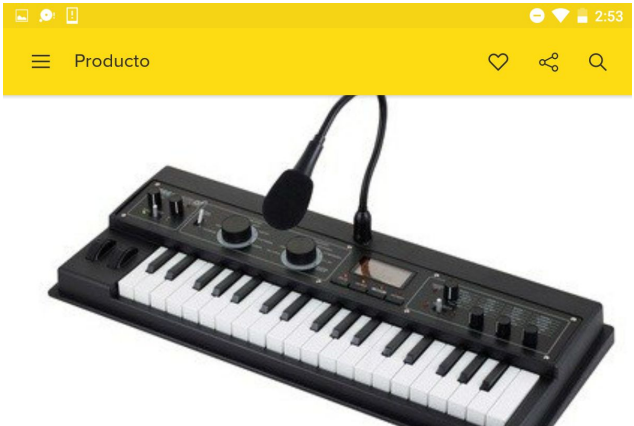
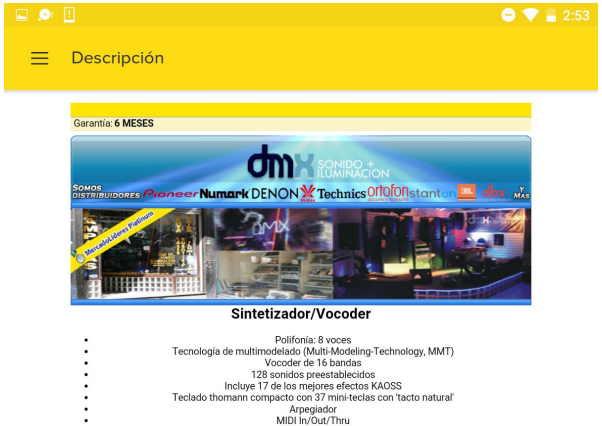
App Bar

La AppBar (también conocido como Action Bar) es la barra que se muestra en la parte superior de nuestras aplicaciones.







Estructura

¿Cómo lo uso?

Digital House > Coding School

1) Definir en el manifest algún tema que diga “NoActionBar”:

Ejemplo

```
android:theme="@style/Theme.AppCompat.Light.NoActionBar">
```

2) Agregar el AppBar al layout del activity. Al elemento AppBar android lo conoce como ToolBar.

```
<android.support.v7.widget.Toolbar
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="t://schemas.android.com/apk/res-auto"
  android:id="@+id/my_toolbar"
  android:layout_width="match_parent"
  android:layout_height="60dp"
  android:background="algun color"
  app:titleTextColor="#ffffff"
 />
```

¿Cómo lo uso?

Digital House > Coding School

3) En el onCreate se le indica al activity que setee su toolbar.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
...
   Toolbar myToolbar = (Toolbar) findViewById(R.id.my_toolbar);
   setSupportActionBar(myToolbar);
...
}
```

Modificar título por defecto

En el manifest podemos definir una etiqueta para el título de nuestro tool bar

```
<activity
    android:name="com.digitalhouse.listview.ListViewActivity"
    android:label="@string/app_bar_movie_list_activity_title">
```

¿Cómo agregar acciones?

Digital House > Coding School

1) Creo un xml en la carpeta menú, dentro de res

```
<?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_search"
     android:icon="@android:drawable/ic_menu_search"
     android:title="@string/action_search"
     app:showAsAction="ifRoom"/>

  <item android:id="@+id/action_plus"
     android:icon="@android:drawable/btn_plus"
     android:title="@string/action_plus"
     app:showAsAction="never"/>

  <item android:id="@+id/action_settings"
     android:title="@string/action_settings"
     android:orderInCategory="100"
     app:showAsAction="never" />
</menu>
```

¿Cómo agregar acciones?

Digital House > Coding School

2) Se sobreescribe el método onCreateOptionsMenu del correspondiente activity

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

¿Cómo se responde a las acciones?

Digital House > Coding School

1) Se sobreescribe el método onOptionsItemSelected

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
   switch (item.getItemId()) {
      case R.id.action_plus:
         return true;
      case R.id.action_search:
         return true;
      case R.id.action_settings:
         return true;
      default:
         return super.onOptionsItemSelected(item);
   }
}
```

Actions bajo la lupa

Digital House > Coding School

```
<item
    android:id="@+id/action_search"
    android:icon="@android:drawable/ic_menu_search"
    android:title="@string/action_search"
    app:showAsAction="ifRoom"/>
```

OrderInCategory
Es el orden de izquierda a derecha que van a aparecer. El 1 tiene precedencia del 2.

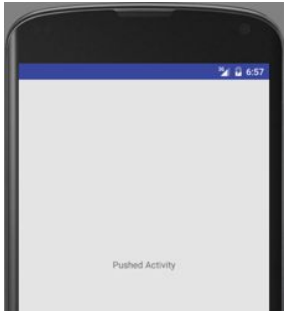
showAsAction

- withText: Muestra el icono mas el texto.
- never: Lo muestra siempre dentro del menu overflow.
- always: Trata de mostrarlo siempre
- ifRoom: Lo muestra si es que hay suficiente espacio.
- collapseActionView: algo mas avanzado que vamos a ver mas adelante

Usar el App bar como navegación

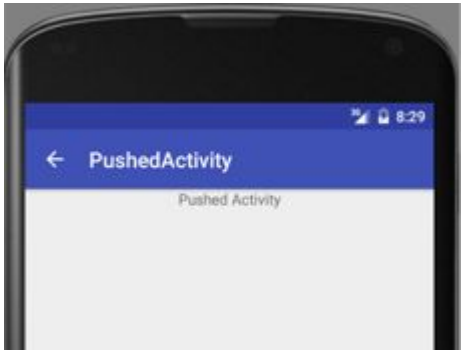
La aplicación de uno debería de hacerle fácil a el usuario volver a la acción anterior a la que estaba realizando. Como buena practica se suele agregar un Up button en el App Bar para que el usuario pueda retornar a lo anterior que estaba haciendo.

Si agregamos otro activity, PushedActivity, le seseamos el titulo en el manifest y lo llamamos, nosotros querríamos que aparezca el UpButton. Sin embargo cuando llamamos a un nuevo activity pasa lo siguiente.



Directamente no aparece el App Bar...Porque?

Y nosotros queriamos esto--->



Usar el App bar como navegación

Digital House > Coding School

Porque el AppBar como dijimos esta relacionado con cada activity, por lo cual, si queremos que este activity tenga un App Bar, vamos a tener que agregarlo en su layout y desde el activity debemos decirle que lo use, así como hicimos la primera vez.
Entonces como seguimos?
Copiamos y pegamos el app bar?.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        tools:context="com.example.digitalhouse.toolbar.MainActivity">
        <android.support.v7.widget.Toolbar
                android:id="@+id/my_toolbar"
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize"
                android:background="?attr/colorPrimary"
                android:elevation="4dp"
                android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar" />
        <Button android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Hello World!"
                android:onClick="pushActivity" />
</LinearLayout>
```


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        tools:context="com.example.digitalhouse.toolbar.PushedActivity">
        <android.support.v7.widget.Toolbar
                android:id="@+id/my_toolbar"
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize"
                android:background="?attr/colorPrimary"
                android:elevation="4dp"
                android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar" />
</LinearLayout>
```

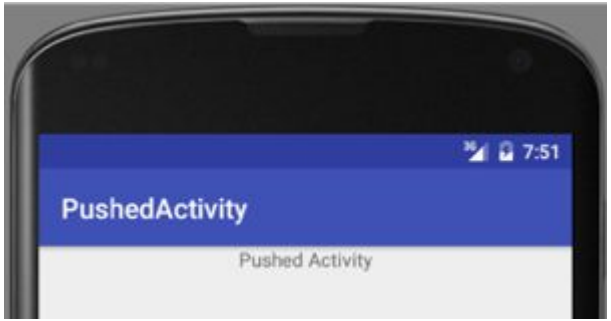
Vamos a incluir el mismo App bar, por lo cual vamos a aprender a como rehusar un layout.
Para hacer esto hay que crear un xml en el layout que solo incluya lo que queremos reusar, en este caso el AppBar.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.Toolbar xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/my_toolbar"
  android:layout_width="match_parent"
  android:layout_height="?attr/actionBarSize"
  android:background="?attr/colorPrimary"
  android:elevation="4dp"
  android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar" />
```

Y lo llamo desde el otro layout de la siguiente manera

```
<include layout="@layout/app_bar"/>
```

Ahora nos aparece el AppBar :)



Y el UpButton?.
Lo tenemos que configurar.

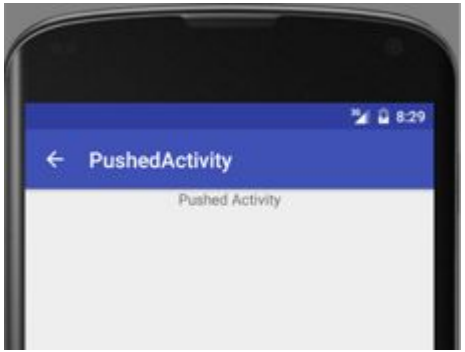
Primero, aunque no parezca logico, tenemos que indicarle a un Activity, cual es su padre activity.
Donde puede ser que tengamos que ponerlo?

En el manifest.
Se le agrega el siguiente atributo al activity a ser instanciado.

android:parentActivityName=“com.example.digitalhouse.toolbar.MainActivity"

Segundo
Decirle al supportActionBar que lo muestre

```
ActionBar ab = getSupportActionBar();
ab.setDisplayHomeAsUpEnabled(true);
```



Hace falta hacer el método onClick de la flechita?.

Ejercicio

Digital House > Coding School

Al hacer click en el botón llama al SecondaryActivity, y desde el primero si seleccionamos la lupa imprime un toast.