

# Elements Avançats de l'IU

## Navigation Drawer

Programació Multimèdia i Dispositius Mòbils



2n. DAM  
2015-2016



# Navigation Drawer

- Què és?
- Navigation Drawer amb Material Design
- Estructura
- Com s'implementa?
- Gestionant Events
- Material Design
- Ara integrant-ho amb Fragments!!!



# Navigation Drawer



<https://developer.android.com/intl/es/training/implementing-navigation/nav-drawer.html>

- Què és?

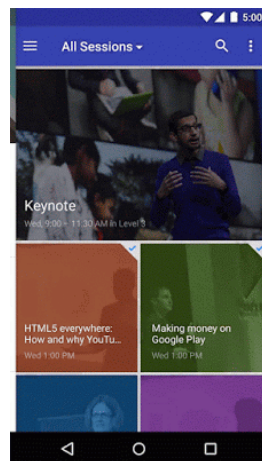
És un **panell** que mostra les principals **opcions de navegació** per a moure'ns dins la nostra app.

Sol estar amagat a la part dreta (o esquerra) de la nostra pantalla.

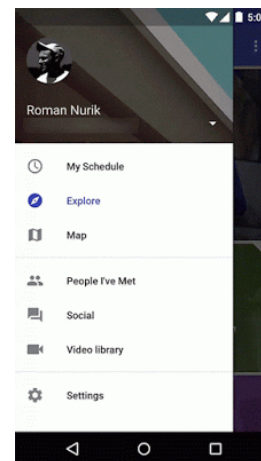
Es mostra quan l'usuari:

- desplaça el dit d'esquerra a dreta des del cantó esquerre de la pantalla.
- polsa sobre l'icona de l'app, a l'ActionBar.

Navigation Drawer tancat



Navigation Drawer desplegat



# Nav. Drawer amb Material Design



El drawer apareix **davant** de l'Action Bar.

La resta de la pantalla apareix semitransparent (enfosquida)

La icona de Navegació de l'Action Bar (ens permet desplegar el Navigation Drawer), queda oculta quan aquest es desplega. En versions anteriors a Lollipop, quedava visible.



El Drawer té una amplada estàndard: No superior a 320 dp en mòbils ni a 400dp en tablets. Ni inferior a l'amplada de la pantalla menys l'amplada de l'Action Bar. (En Nexus 5,  $360dp - 56dp = 340dp$ )



# Nav. Drawer amb Material Design



L'alçada dels ítems, en un Nav. Drawer:

- 48 dp d'alçada de cada fila
- 8 dp per sobre i per sota dels divisors
- 8 dp anteriors a la llista de seccions

Més informació i detalls sobre el disseny del Navigation Drawer en:

<https://www.google.com/design/spec/patterns/navigation-drawer.html>





# Estructura

- Es divideix en dues parts:

Ho definirem TOT  
en un fitxer XML



Generalment el  
View per  
mostrar les  
distintes opcions  
és un **ListView** o  
un **GridView**

El ListView / GridView portarà associats altres fitxers XML i JAVA per poder funcionar





# Com s'implementa?

## Requeriments:

- Biblioteca de compatibilitat **v4** d'Android

Obrirem l'script build.gradle del nostre Mòdul i afegirem a l'apartat de dependències la següent línia:

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    compile 'com.android.support:support-v4:22.1.0'  
}
```

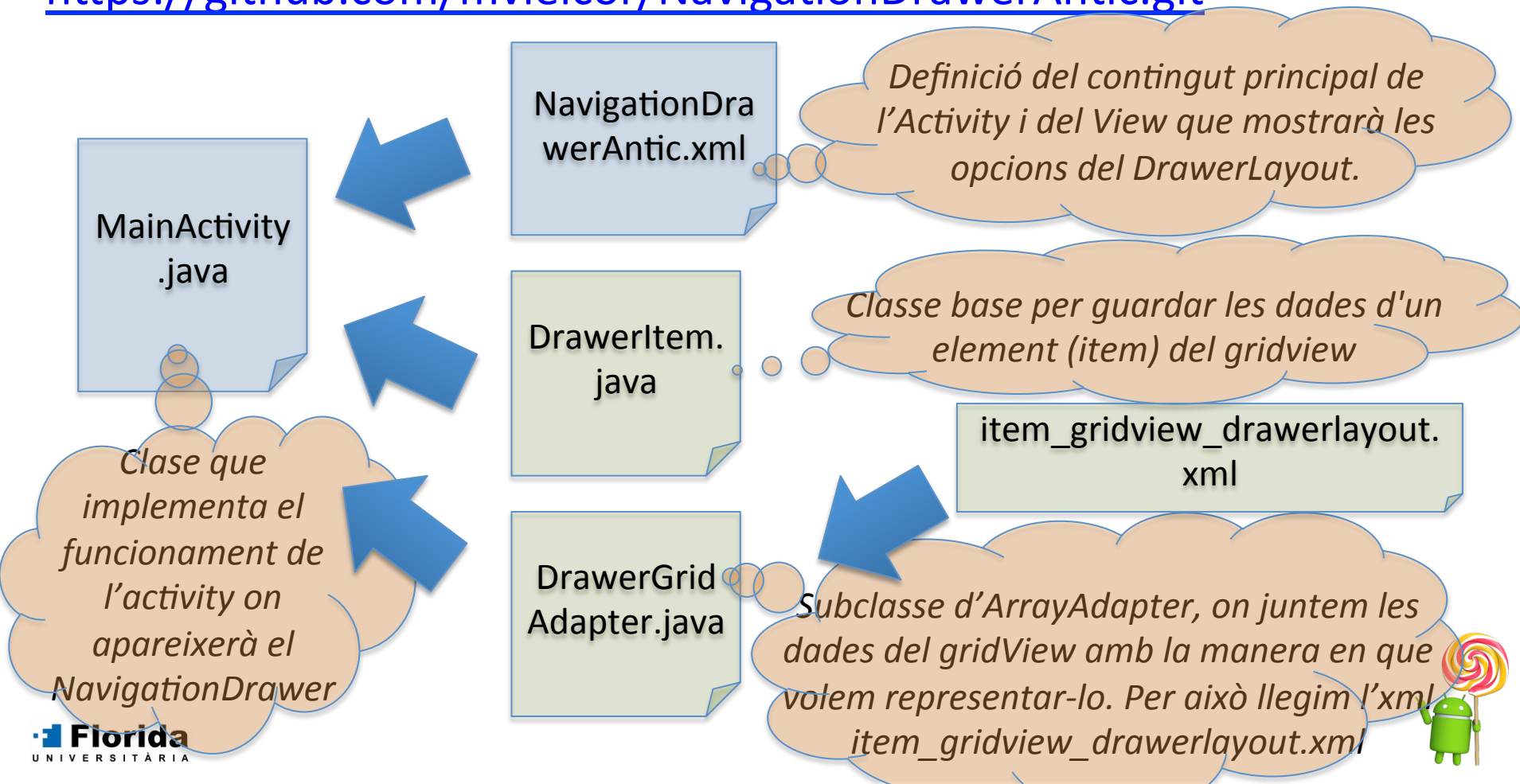




# Com s'implementa?

Descarregueu el projecte

<https://github.com/mvielcor/NavigationDrawerAntic.git>







# Com s'implementa?

NavigationDrawerAntic.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/elDrawerLayout">
```

Utilitzarem el View  
DrawerLayout, que  
sínclou dins la  
biblioteca de suport v4.

Li posarem un  
identificador, per  
poder-lo referenciar  
després des de  
l'Activity que mostrarà  
el Drawer Layout.

En este cas, les opcions  
les mostrarem en un  
GridView de 2  
columnes .

*<!-- 1a part. En este cas, ho organitzarem tot mitjaçant un Relative Layout -->*

```
<RelativeLayout
    android:id="@+id/elNavigationDrawerAntic"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView android:text="@string/contingut_principal" android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true" />

</RelativeLayout>
```

*<!-- 2a part. Utilitzarem un GridView per a mostrar les opcions del Navigation Drawer -->*

```
<GridView
    android:id="@+id/grid_Opcions_Nav_Drawer"
    android:layout_width="240dp"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:scrollbars="vertical"
    android:numColumns="2"
    android:verticalSpacing="10dp"
    android:horizontalSpacing="10dp"
    android:stretchMode="columnWidth"
    android:background="#ffffff"
    android:gravity="center"/>
```

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





# Com s'implementa?

**MainActivity.java**

Carreguem el contingut  
del fitxer  
*NavigationDrawerAntic.xml*

Creem un objecte  
mDrawerLayout que  
referencia el que hem  
definit a l'XML

Creem un Objecte  
GridView que  
referencia el creat a  
l'xml.

Establim un Adaptador  
per a carregar el  
GridView amb les  
distintes opcions

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

    // Instanciem el DrawerLayout
    mDrawerLayout = (DrawerLayout) findViewById(R.id.elDrawlerLayout);
    mGridViewDrawer = (GridView) findViewById(R.id.grid_Opcions_Nav_Drawer);

    // Creem un arrayList d'objectes DrawerItem
    for(int i=0;i<mTEXTOS_OPCIONS.length;i++){
        LlistItems.add(new DrawerItem(mTEXTOS_OPCIONS[i],mICONS[i]));
    }

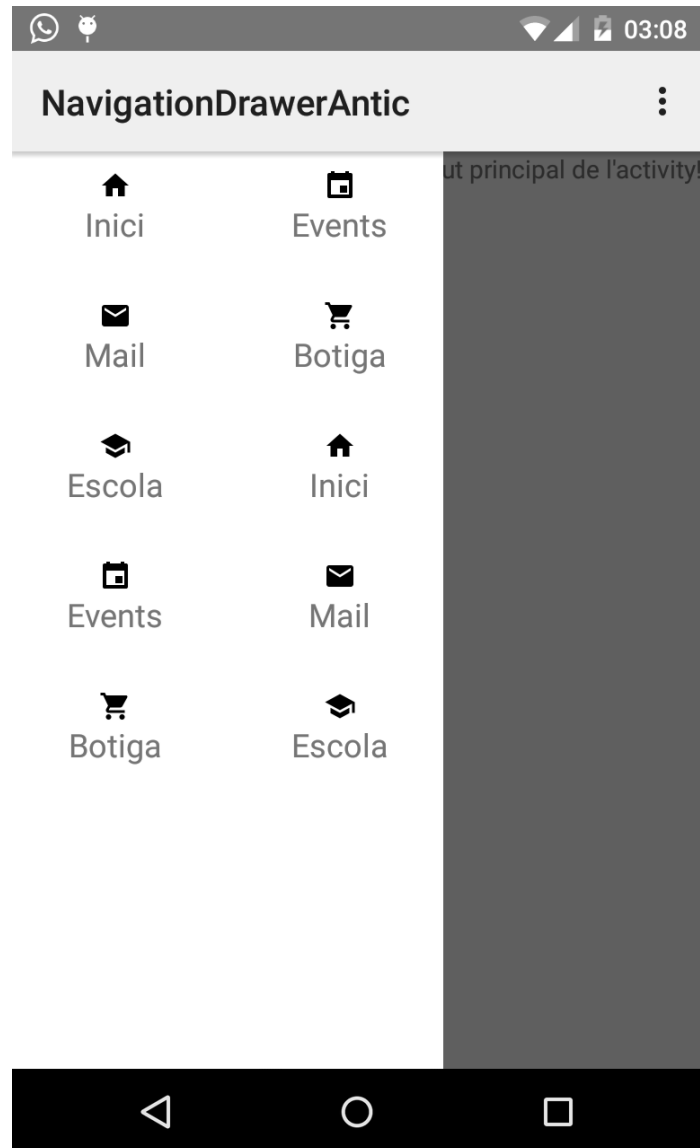
    // Establim l'Adaptador del GridView
    mGridViewDrawer.setAdapter(new DrawerGridAdapter(this,LlistItems));
}
```

Hem obviat, intencionadament, la manera de crear  
l'Adaptador i de passar-li la informació a cada ítem del  
GidView.

Consulteu el projecte per vore el codi complet.



# El Resultat





# Gestionant Events

## Dins la llista del NavigationDrawer

- Necessitem afegit un Listener al navigation Drawer, **OnItemClickListener** i implementar els seus mètodes :

*// Afegim un Listener per a que reaccione a la nostra selecció*

```
mGridViewDrawer.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    public void onItemClick(AdapterView<?> parent, View v, int position, long id) {  
        // Accedim al Linear Layout definit en item_gridview_drawerlayout.xml  
        LinearLayout ll = (LinearLayout) v;  
        //accedim a l'Element TextView que ocupa la segona posició  
        TextView tv = (TextView) ll.getChildAt(1);  
        //Mostrem un Toast amb el text i la posició que ocupa l'ítem seleccionat dins del gridview  
        Toast.makeText(getApplicationContext(),tv.getText()+ " (" + position+")",  
            Toast.LENGTH_SHORT).show();  
    }  
});
```





# Gestionant Events

## Obrint i Tancant el NavigationDrawer

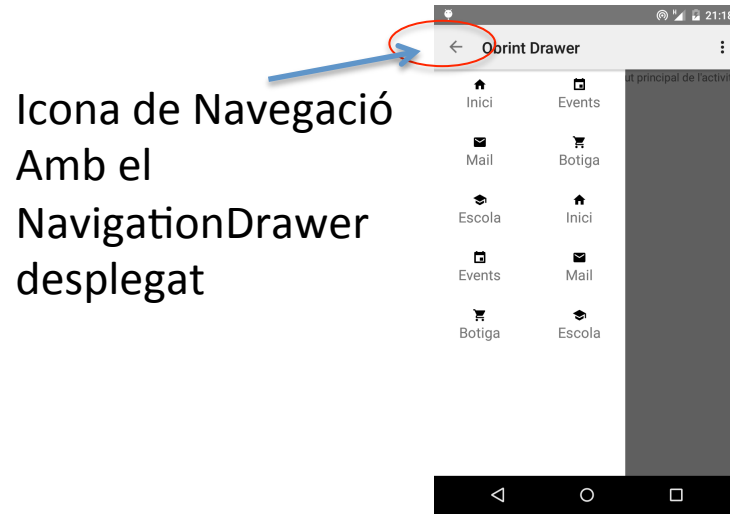
- Classe **ActionBarDrawerToggle**

Aquesta classe proporciona una forma pràctica d'unir la funcionalitat del DrawerLayout i la Barra d'accions per a implementar el disseny recomanat dels NavigationDrawer. Cal afegir la biblioteca de suport v7. Al Gradle afegirem la línia

**compile 'com.android.support:appcompat-v7:22.0.+'**



Icona de Navegació  
(icona d'hamburguesa)



Icona de Navegació  
Amb el  
NavigationDrawer  
desplegat





# Gestionant Events

## Obrint i Tancant el NavigationDrawer

### 4 Pasos a seguir:

1. mostrar la icona de navegació a la Barra d'Accions
2. Crear i implementar l'objecte ActionBarDrawerToggle
3. Associar-lo com a Listener del NavigationDrawer
4. Gestionar els clics de quan es polsa la icona de Navegació

---

PAS 1     *// Mostrem la icona de navegació a l'ActionBar*  
          `getActionBar().setDisplayHomeAsUpEnabled(true);`  
          *// Activem l'icona de navegació de l'ActionBar per a que pugui*  
          *// mostrar i ocultar el navigationDrawer*  
          `getActionBar().setHomeButtonEnabled(true);`





# Gestionant Events

## Obrint i Tancant el NavigationDrawer

PAS 2

```
mDrawerToggle = new ActionBarDrawerToggle(this, mDrawerLayout,R.string.openDrawer,
R.string.closeDrawer) {
    @Override
    public void onDrawerOpened(View drawerView) {
        super.onDrawerOpened(drawerView);
        // Ací posarem el codi que vulguem que s'execute quan el navigationDrawer esta obert
        //Per fer una prova, li direm que mostre el text "Obrint Drawer"
        getActionBar().setTitle(R.string.openDrawer);
        //Sincronitzem (important)
        mDrawerToggle.syncState();
    }
    @Override
    public void onDrawerClosed(View drawerView) {
        super.onDrawerClosed(drawerView);
        // Ací posarem el codi que vulguem que s'execute quan el navigationDrawer esta obert
        //Per fer una prova, li direm que mostre el text "Tancant Drawer"
        getActionBar().setTitle(R.string.closeDrawer);
        //Sincronitzem (important)
        mDrawerToggle.syncState();
    }
};
```





# Gestionant Events

## Obrint i Tancant el NavigationDrawer

### PAS 3

```
// Li associem com a Listener, el l stener del ActionBar  
// Listener que escolta quan s'obri o tanca  
// el NavigationDrawer des del toggle de l'ActionBar  
mDrawerLayout.setDrawerListener(mDrawerToggle);  
mDrawerToggle.syncState();           // Finalment, sincronitzem l'estat del toggle.
```

### PAS 4

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {
```

```
    // Gestionem els Clicks de l'ActionBar ac . L'Action bar gestionar  els clicks de la icona Home/Up
```

```
    // Si s'ha pulsat L'icona de l'ActionBar, tornarem true, per indicar que hem gestionat l'event
```

```
    if (mDrawerToggle.onOptionsItemSelected(item)) {
```

```
        return true;
```

```
    }
```

```
    // En qualsevol altre cas, tornarem super.onOptionsItemSelected(item) per passar l'event a
```

```
    // la classe pare, i que el gestione ella.
```

```
    return super.onOptionsItemSelected(item); }
```





# Material Design



Però si us fixeu, no es compleix la norma 1 del Material Design (que recomana Google) per al navigation Drawer “El drawer apareix **davant** de l’Action Bar”

Per a aconseguir-ho, només ens cal:

1. Utilitzar la Support Library v7 (que ja ho feiem).
2. Canviar el nostre Activity per un ActionBarActivity
3. Assegurar-nos d'utilitzar un tema sense ActionBar (utilitzarem el nou Toolbar)

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

```
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar"> Styles.xml
```



# Material Design.



## 4. Definir un objecte ToolBar en un layout xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.Toolbar xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/ColorPrincipal"
    android:theme="@style/ThemeOverlay.AppCompat.Dark"
    android:elevation="4dp"
    android:paddingTop="5dp">
</android.support.v7.widget.Toolbar>
```

Fitxer: tool\_bar.xml

## 5. Incloure eixe xml com a part del contingut principal de l'activity, en l'xml del DrawerLayout

```
<!-- 1a part. En este cas, ho organitzarem tot mitjançant un Relative Layout -->
<RelativeLayout
    <include
        android:id="@+id/tool_bar"
        layout="@layout/tool_bar">
    </include>
    ....
```

Fitxer: navigation\_drawer\_antic.xml



# Material Design.



6. A l'Activity que carrega el NavigationDrawer:

- Heretarà de **ActionBarActivity**, en compte d'**Activity**  
*public class MainActivity extends ActionBarActivity {...}*
- Crearem un atribut de classe que serà un objecte de la classe **ToolBar**  
*private Toolbar toolbar;*
- Instanciem l'objecte toolbar amb l'element de l'xml creat al punt 4
- Establim eixe objecte com la nova ActionBar: *setSupportActionBar(toolbar);*
- Canviem totes les intàncies **getActionBar()** per *getSupportActionBar()*

```
toolbar = (Toolbar) findViewById(R.id.tool_bar);
if (toolbar!=null) {
    setSupportActionBar(toolbar);
    // Mostrem la icona de navegació a l'ActionBar
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    // Activem l'icona de navegació de l'ActionBar per a que puga mostrar i ocultar el navigationDrawer
    getSupportActionBar().setHomeButtonEnabled(true);
}
```

Fitxer: MainActivity.java

