

Día 2 (Introducción a Material Design y AppCompatActivity)

1. Agregar dependencias al proyecto

- a. En el archivo **build.gradle** del módulo app se debe agregar la dependencia hacia la biblioteca de soporte de diseño:

- i. `compile 'com.android.support:design:24.2.1'`

2. Crear MainActivity

- a. Crear la clase **MainActivity** que hereda de [android.support.v7.app.AppCompatActivity].
- b. **MainActivity** debe estar declarada dentro del **AndroidManifest.xml**.
- c. Crear un **layout (xml)** que se muestre en el onCreate() de MainActivity. Este layout contiene la siguiente jerarquía de views:
 - i. DrawerLayout
 1. CoordinatorLayout
 - a. AppBarLayout
 - i. Toolbar
 - b. FrameLayout [Contenedor para Fragments]
 2. NavigationView

3. Crear ActionBarDrawerToggle

- a. Se debe configurar el **toolbar** del layout del **MainActivity** como el default de la Activity.
- b. Se debe **habilitar** el icono de **home** en el toolbar.
- c. Se debe crear un **ActionBarDrawerToggle** y agregar como listener del **DrawerLayout**.
- d. Se deben sobrescribir los métodos MainActivity#onConfigurationChanged() y MainActivity#onOptionsItemSelected() y llamar los métodos correspondientes del ActionBarDrawerToggle.
- e. Se debe sobrescribir el método MainActivity#onPostCreate() y llamar el método ActionBarDrawerToggle#syncState()

4. Crear nuevo estilo para eliminar ActionBar del sistema

- a. Se debe crear un estilo en el archivo **res/values/styles.xml** que oculte el action bar del sistema. Este estilo se le aplica al **MainActivity** dentro del AndroidManifest con la propiedad **android:theme**.

5. Agregar los elementos al NavigationView

- a. Crear el folder res/menu
- b. Crear un xml dentro de res/menu que contenga los ítems a mostrar. Importante cada ítem debe tener un **id**.
- c. Dentro del layout (xml) del MainActivity al elemento **NavigationView** se le debe configurar el atributo **app:menu**.

6. Reaccionar a clicks en el menu

- a. Esto se logra llamando el método *NavigationView#setNavigationItemSelectedListener()* en el método *onCreate()* de **MainActivity** después de haber llamado *setContentview(int)*.
- b. Dentro del método callback se puede agregar/reemplazar fragments en el view contenedor que se agregó al layout.

7. Crear TasksFragment

- a. Crear la clase **TasksFragment** que hereda de [android.support.v4.app.Fragment].
- b. Crear un **layout (xml)** que se muestre en el *onCreateView()* de TasksFragment. Este layout contiene contenido Dummy solo para verificar que el Fragment se muestre en pantalla.

8. Ejercicios adicionales

- a. Agregar un **header** al NavigationView. Como referencia pueden crear un nuevo proyecto en Android Studio y seleccionar Navigation Drawer Activity como Activity inicial.