

# Laboratorio 7.- Gestión de contactos

---

## Contenido:

1	NAVIGATION DRAWER.....	2
2	APLICACIÓN A REALIZAR: GESTIÓN DE CONTACTOS.....	5

**Objetivos:** aprender la definición y el uso de los menús desplegados y adquirir destreza en el uso de los proveedores de contenido añadiendo, modificando y eliminando contenido.

# 1 Navigation Drawer

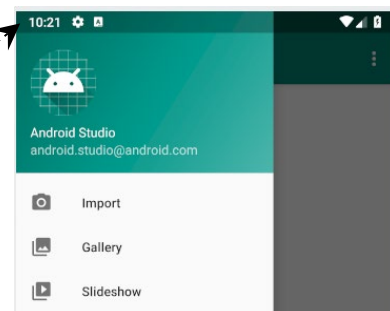
El Navigation Drawer es el menú lateral deslizante (ver Fig. 2) que tienen muchas aplicaciones Android. Siguiendo las directrices establecidas por Material Design, este menú debe estar oculto en la parte izquierda y desplegarse al deslizar el dedo horizontalmente desde el borde izquierdo del dispositivo al lado derecho. También debe existir un icono en la Action Bar que despliegue el menú lateral (ver Fig. 1). Al desplegar el menú lateral, este debe quedar situado por encima de la Action Bar, pero por debajo de la barra de estado (donde se muestran las notificaciones, el nivel de batería, etc.)



Fig. 1.- Menú lateral recogido

Icono para desplegar

Status Bar



Action Bar

Fig. 2.- Menú lateral desplegado

El menú lateral se crea para una actividad, no para toda la aplicación. Si se quiere usar el menú lateral en más de una actividad, se puede hacer que todas las actividades que lo necesiten hereden de la actividad que tiene el menú lateral. Otra posibilidad es usar la actividad que tiene el menú lateral como un contenedor de Fragments, donde se irán intercambiando los Fragments y por lo tanto, la funcionalidad, pero la actividad (el contenedor) será todo el rato la misma.

Para utilizar un Navigation Drawer lo primero que hay que hacer es incluir las siguientes dependencias en nuestra aplicación (en el fichero Gradle):

```
implementation 'androidx.drawerlayout:drawerlayout:1.1.1'
implementation 'androidx.navigation:navigation-ui:2.3.4'
```

A continuación, hay que modificar el layout correspondiente a la actividad para que su elemento raíz sea <DrawerLayout> y para que en su interior, además de los elementos gráficos correspondientes a la actividad, contenga también un elemento de tipo <NavigationView>.

```
<androidx.drawerlayout.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/drawer_layout"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:fitsSystemWindows="true"
tools:openDrawer="start">

  <include
    layout="@layout/contenido"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

  <com.google.android.material.navigation.NavigationView
    ... />
</androidx.drawerlayout.widget.DrawerLayout>
```

Ayuda a organizar el XML. Es equivalente a poner aquí el texto que está en fichero contenido.xml de la carpeta layout. En ese fichero está definida la interfaz de la actividad.

El ID del drawer

El NavigationView siempre el último para que, al desplegarlo, quede por encima.

## Desarrollo Avanzado de Software - Curso 2023 / 2024

Como las normas de Material Design dicen que en la barra de acciones debe existir un icono para desplegar y cerrar el menú lateral, hay que quitar la barra de acciones que se incluye por defecto y cambiarla por una Toolbar tal y como se describía en el Laboratorio 5.

El elemento «NavigationView» tiene dos partes importantes y que se definen a través de dos ficheros XML: la cabecera (atributo headerLayout) y el menú (atributo menu).

Se ajusta a la ventana del sistema y se mostrará por debajo de la barra de estado

```
<com.google.android.material.navigation.NavigationView
    android:id="@+id/elnavigationview"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:fitsSystemWindows="true"
    app:headerLayout="@layout/lacabecera"
    app:menu="@menu/lasopciones" />
```

En qué lado "reposa" el menú. start (izquierda) o end (derecha)

La cabecera es la parte superior del menú desplegable y el menú son las distintas opciones que figuran en el menú desplegable (ver Fig. 3)

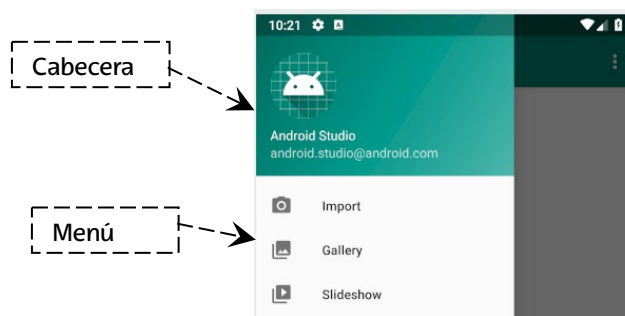


Fig. 3.- Cabecera y menú del menú desplegable

El fichero xml correspondiente a la cabecera es un fichero xml de layout normal y corriente donde se puede usar el layout que se desee (linear, grid, constraint, etc.) y los elementos gráficos que se quiera. Este fichero se almacena en la carpeta layout.

El fichero xml correspondiente a las opciones del menú se almacena en la carpeta menú. La raíz del fichero xml debe ser el elemento «menu» y cada opción del menú se define mediante el elemento «item». A su vez, un «item» puede contener como elemento un «menu» para crear submenús. Las opciones del menú también se pueden agrupar mediante el elemento «group» para aplicar alguna característica, como por ejemplo visibilidad, de manera simultánea a todas las opciones que estén incluidas en el grupo.

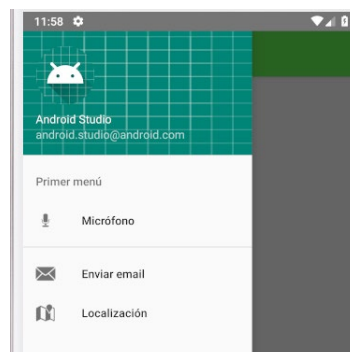


Fig. 4.- Ejemplo de menú

## Desarrollo Avanzado de Software - Curso 2023 / 2024

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:showIn="navigation_view">

    <item android:title="Primer menú">
        <menu>
            <item
                android:id="@+id/microfono"
                android:icon="@android:drawable/ic_btn_speak_now"
                android:title="Micrófono" />
        </menu>
    </item>
    <item
        android:id="@+id/escribir"
        android:icon="@android:drawable/ic_dialog_email"
        android:title="Enviar email" />
    <item
        android:id="@+id/localizacion"
        android:icon="@android:drawable/ic_dialog_map"
        android:title="Localización" />
</menu>
```

Tras la definición de la interfaz gráfica correspondiente a la actividad, a la cabecera y a las opciones del menú se pasa a trabajar con el código.

Por un lado, estableceremos como barra de acción la que hayamos definido en el layout de nuestra actividad (ver Laboratorio 5).

Por otro lado, en el `onCreate` de la actividad también se recoge el objeto `NavigationDrawer` y el objeto `NavigationView`. Sobre este último se define un listener para poder reaccionar a la interacción del usuario sobre las opciones del menú.

```
final DrawerLayout elmenudesplegable = findViewById(R.id.drawer_layout);
NavigationView elnavigation = findViewById(R.id.nav_view);
elnavigation.setNavigationItemSelectedListener(new
    NavigationView.OnNavigationItemSelectedListener() {
        @Override
        public boolean onNavigationItemSelected(@NonNull MenuItem item) {
            switch (item.getItemId()) {
                case R.id.microfono:
                    ...
                    break;
                case R.id.localizacion:
                    ...
                    break;
            }
            elmenudesplegable.closeDrawers();
            return false;
        }
    });
```

Qué hacer si el usuario pulsa la opción del menú llamada micrófono

Plegar el menú

Para completar las recomendaciones de Material Design, debemos añadir a nuestra barra de acción un icono en el lado izquierdo que indique la existencia de un menú (ver Fig. 5) y que al pulsarlo despliegue el menú lateral. Además, el menú debe ocultarse si al estar desplegado se pulsa el botón Back del dispositivo.

Para incluir el icono en la barra de acción sólo hay que indicarle qué icono queremos que se muestre como opción "Home" y activarlo.

Tras haber sustituido la barra de acciones por la nuestra

```
getSupportActionBar().setHomeAsUpIndicator(android.R.drawable.ic_dialog_info);
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
```

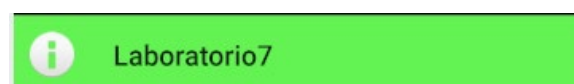


Fig. 5.- Icono Home

## Desarrollo Avanzado de Software - Curso 2023 / 2024

La opción "Home" es una opción que incluye Android por defecto como una opción del menú, por lo que la reacción a la interacción del usuario con dicha opción, se trata como una opción más del menú de la barra de acción (ver Laboratorio 5).

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

    switch(item.getItemId()) {
        case android.R.id.home:
            elmenudesplegable.openDrawer(GravityCompat.START);
            return true;
        ...
    }

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

Para que el menú desplegable se cierre al pulsar el botón Back, hay que sobrescribir el método `onBackPressed()` para que si el menú está abierto, lo cierre. Si el menú no está abierto, se llama al método superior para que tenga el comportamiento que corresponda.

```
@Override
public void onBackPressed() {
    if (elmenudesplegable.isDrawerOpen(GravityCompat.START)) {
        elmenudesplegable.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
}
```

## 2 Aplicación a realizar: gestión de contactos

Cread una aplicación que tenga un menú desplegable con 3 opciones: Añadir, Modificar y Borrar.

Si el usuario elige la opción de Añadir se le mostrará una interfaz para que introduzca los datos para añadir un nuevo contacto al dispositivo.

Si el usuario elige la opción de Modificar, se le mostrará una interfaz con un listado de todos los contactos que tiene en el dispositivo y tras seleccionar uno, podrá modificar sus datos.

Si el usuario elige la opción de Borrar, se le mostrará una interfaz con un listado de todos los contactos que tiene en el dispositivo y tras seleccionar uno, podrá eliminarlo.