

Tutorial. Bottom navigation + Fragments

Objectiu

Crear un menú de navegació amb fragments

Demo

Descarrega la demo [aquí](#)

Pas a pas.

MENÚ:

- Crearem els 3 vector assets amb els icones que vulguem mostrar al menú
 - Home
 - Llistat
 - Formulari
- Crearem una carpeta per gestionar el menú
 - res → new → Android resource directory.
 - Resource type: menu
 - Directory name: menu
- Dins de menu farem crearem un nou menu resource file
 - res → menu → botó dret → new → Menu resource file
 - Name: bottom_navigation
- Obrirem l'XML generat i afegirem els 3 ítems mitjançant el codi xml, especificant l'id, el títol i la icona.

```
<item android:id="@+id/nav_home"  
      android:title="Home"  
      android:icon="@drawable/ic_home"  
/>
```

ACTIVITY_MAIN (XML):

- A l'activity on vulguem posar el menú afegirem el BottomNavigationView
 - id: main_menu
 - width: match_parent
 - height: wrap_content
 - app:menu el menú que hem creat anteriorment (app:menu="@menu/bottom_navigation")
- A l'activity afegirem el FrameLayout per poder visualitzar els diferents fragments
 - id: fragment_contaner

- Fixa el top a la part superior de la pantalla, left i right als extrems i el bottom l'has de fixar amb el top del menú
- width: match_parent
- height: match_parent

FRAGMENTS:

- A la carpeta RES crearem els diferents fragments, per a fer-ho tenim dos mètodes diferents:
 - Fer una banda new → Fragment → Blank Fragment. Això ens generarà un XML i un JAVA que ja aniran vinculats. Si ho feu, veureu que al JAVA ja hi ha molts mètodes implementats que ara per ara no necessitem. Els podem deixar i fixar-nos només amb el mètode `onCreateView`
 - D'altra banda podem crear un new → Layout Resource File i un new → Java Class i posar-hi com a SuperClass Fragment ja que hereta d'aquesta classe. Finalment hauréu de sobreesciure el mètode `onCreateView` que serà el que ens mostrarà el fragment.

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_home, container,
false);
}
```

MAIN ACTIVITY (JAVA):

- Tornem al main activity per programar el menú i que faci els canvis de pantalla.

```
BottomNavigationView bottomNav = findViewById(R.id.main_menu);

bottomNav.setOnItemSelectedListener(item -> {
    Fragment selectedFragment = null;
    switch (item.getItemId()) {
        case R.id.nav_home:
            selectedFragment = new FragmentHome();
            break;
    }

    getSupportFragmentManager().beginTransaction().replace(R.id.fragment
nt_container, selectedFragment).commit();

    return true;
});
```

ACTIVITY_MAIN (XML): Si volem que el menú no tingui els labels hem d'utilitzar la propietat `labelVisibilityMode`

```
<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/main_menu"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:labelVisibilityMode="unlabeled"
    app:layout_constraintStart_toStartOf="parent"
    app:menu="@menu/bottom_navigation" />
```