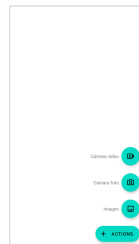


Exercici resol FAB menu

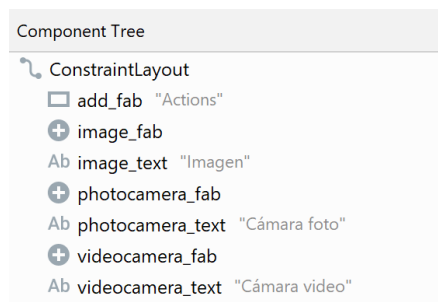
[Descarregar aquests apunts](#)

Quan premem un **FAB** podem aplicar transicions a aquest. És possible que ens servisca per a mostrar un menú que conté un conjunt d'accions (de tres a sis segons les guies de disseny) o una **Toobar** o bé una activitat nova.

Desenvoluparem un menú d'accions vinculat al **FAB** que tindrà el següent aspecte una vegada premut el botó, a més apareixerà un text associat al **FAB** i cadascuna de les seues accions:



Vegem els components necessaris per a la interfície:



L'arxiu xml corresponent a aquesta interfície seria:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context=".MainActivity">
```

```
<com.google.android.material.floatingactionbutton.ExtendedFloatingActionButton  
    android:id="@+id/add_fab"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="end"  
    android:layout_marginEnd="16dp"  
    android:layout_marginBottom="16dp"  
    android:text="Actions"  
    app:icon="@drawable/baseline_add_black_24dp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent" />
```

```
<com.google.android.material.floatingactionbutton.FloatingActionButton  
    android:id="@+id/image_fab"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="24dp"  
    app:fabSize="normal"  
    app:layout_constraintBottom_toTopOf="@id/add_fab"  
    app:layout_constraintEnd_toEndOf="@id/add_fab"  
    app:srcCompat="@drawable/outline_image_24" />
```

```
<TextView  
    android:id="@+id/image_text"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginEnd="8dp"  
    android:text="Imagen"  
    app:layout_constraintBottom_toBottomOf="@id/image_fab"  
    app:layout_constraintEnd_toStartOf="@id/image_fab"  
    app:layout_constraintTop_toTopOf="@id/image_fab" />
```

```
<com.google.android.material.floatingactionbutton.FloatingActionButton  
    android:id="@+id/photocamera_fab"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="24dp"  
    app:fabSize="normal"  
    app:layout_constraintBottom_toTopOf="@id/image_fab"  
    app:layout_constraintEnd_toEndOf="@id/image_fab"  
    app:layout_constraintStart_toStartOf="@id/image_fab"  
    app:srcCompat="@drawable/outline_photo_camera_24" />
```

```

<TextView
    android:id="@+id/photocamera_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:text="Cámara foto"
    app:layout_constraintBottom_toBottomOf="@id/photocamera_fab"
    app:layout_constraintEnd_toStartOf="@id/photocamera_fab"
    app:layout_constraintTop_toTopOf="@id/photocamera_fab" />

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/videocamera_fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="24dp"
    app:fabSize="normal"
    app:layout_constraintBottom_toTopOf="@id/photocamera_fab"
    app:layout_constraintEnd_toEndOf="@id/photocamera_fab"
    app:layout_constraintStart_toStartOf="@id/photocamera_fab"
    app:srcCompat="@drawable/outline_video_camera_front_24" />

<TextView
    android:id="@+id/videocamera_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:text="Cámara video"
    app:layout_constraintBottom_toBottomOf="@id/videocamera_fab"
    app:layout_constraintEnd_toStartOf="@id/videocamera_fab"
    app:layout_constraintTop_toTopOf="@id/videocamera_fab" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Aclaracions:

- **Línia 2:** el element contenidor principal serà un **ConstraintLayout** . Recordar que les vistes incloses en el mateix hauran de portar almenys dos ancoratges que determinen la seua posició.
- **Línies 9-19:** per a poder implementar aquesta funcionalitat utilitzem com element principal un **ExtendedFloatingButton** . En la línia 16 especifiquem el text que es visualitzarà quan el **FAB** estiga premut. La resta de propietats ja estan explicades.
- **Línies 21-29:** ací definim el primer **FAB** del nostre menú d'accions, les propietats són també fàcilment comprensibles. L'ancoratge del mateix es defineix en funció del **FAB** principal.
- **Línies 31-39:** ací definim el text associat a l'acció. La seua posició es defineix en referència a la posició al **FAB** al qual està associat.
- Aquesta estructura de **FAB** i **TextView** es repeteix per a totes les accions que vulguem incorporar.

Solament ens falta analitzar el codi de la nostra aplicació, on anirem visualitzant els elements de la vista i les accions de cadascun dels botons.

```

class MainActivity : AppCompatActivity() {
    2    var isAllFabsVisible: Boolean? = null
    private lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        val view = binding.root
        setContentView(view)
    10    binding.imageFab.visibility= View.GONE
        binding.photocameraFab.visibility= View.GONE
        binding.videocameraFab.visibility= View.GONE
        binding.imageText.visibility= View.GONE
        binding.photocameraText.visibility= View.GONE
    15    binding.videocameraText.visibility= View.GONE
    16    isAllFabsVisible = false

    18    binding.addFab.shrink()

    20    binding.addFab.setOnClickListener(){
        if (isAllFabsVisible==false){
            isAllFabsVisible=true
            binding.imageFab.show()
            binding.photocameraFab.show()
            binding.videocameraFab.show()
            binding.imageText.visibility= View.VISIBLE
            binding.photocameraText.visibility= View.VISIBLE
            binding.videocameraText.visibility= View.VISIBLE
            binding.addFab.extend()
        }
        else{
            isAllFabsVisible=false
            binding.imageFab.hide()
            binding.photocameraFab.hide()
            binding.videocameraFab.hide()
            binding.imageText.visibility= View.GONE
            binding.photocameraText.visibility= View.GONE
            binding.videocameraText.visibility= View.GONE
            binding.addFab.shrink()
        }
    41    }

    43    binding.imageFab.setOnClickListener(){
        Toast.makeText(applicationContext,"Pulsaste el botón IMAGEN",Toast.LENGTH_LONG)
    45    }

    binding.photocameraFab.setOnClickListener(){
        Toast.makeText(applicationContext,"Pulsaste el botón CAMARA",Toast.LENGTH_LONG)
    }

    binding.videocameraFab.setOnClickListener(){

```

```

        Toast.makeText(applicationContext, "Pulsaste el botón VIDEO", Toast.LENGTH_LONG)
    }
}
}

```

Aclaracions:

- **Línia 2:** utilitzem la variable `isAllFabsVisible` per a saber si està estès o no el **FAB** principal, en principi (línia 16) s'inicia a `false` indicant que està el menú contret.
- **Línies 10-15:** les vistes associades al menú d'accions estan no visibles, per a això utilitzem l'atribut `visibility` de cadascuna d'elles i li associem el valor `View.GONE` que fa que la vista siga invisible sense que la vista ocupe espai en el disseny. `View.INVISIBLE` fa que la vista siga invisible però ocupant espai.
- **Línies 18** amb el mètode `shrink()` es desactiva el mode extendido del **FAB** ocultant el text que té associat quan s'expandisca.
- **Línies 20-41:** ací gestionem el `clik` sobre el botó principal. Si el menú d'accions estava invisible, llavors fem visibles les vistes amb `View.VISIBLE` i fem que aparega el text associat al **FAB** inicial amb el mètode `extend()`. Per contra si el **FAB** principal estava premut, apliquem la lògica inversa a l'explicat anteriorment. El mètode `hide()` oculta la vista amb animació. Similar a `show()` que la mostra amb animació.
- **Línies 43-45** i següents, on s'implementa el `clik` de cadascuna de les accions del menú. Simplement visualitzem un **Toast** indicant l'acció premuda.