SCROLLING ACTIVITY

Este es un tipo de Activity que nos permite diseñar vistas de detalle con una visualización más propia y detallada, por tanto, suele estar reservada para visualizaciones y detalles de objetos sobre los que queremos mostrar toda la información.

Suelen venir de la navegación de una lista de elementos y al seleccionar uno de ellos nos aparece esta vista de detalle.

Creamos un proyecto con una *Scrolling Activity*

Si ejecutamos la aplicación vemos que aparece un espacio donde cargaremos una imagen, y al hacer scroll, la imagen se minimiza y podemos bajar para leer el texto.

Vamos a acceder al layout *activity_scrolling* y lo vamos a personalizar

Primero, entre el elemento *CollapsinToolbar*, que es el que se encarga de hacer la acción de contraer el Toolbar, y el propio *Toolbar*, que hay justo debajo, vamos a incluir una imagen (ImageView)

A esta imagen le vamos a asignar un identificador y dos propiedades:

- fitsSystemWindows=”true”: para que la imagen se ajuste a la ventana del dispositivo
- layout_collapseMode=”parallax”: para que al subir el scroll, la imagen suba. Otra opción sería “pin”, para que se desplace la imagen a la vez que hace scroll

```
<android.support.design.widget.CollapsingToolbarLayout
    android:id="@+id/toolbar_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    app:contentScrim="?attr/colorPrimary"
    app:layout_scrollFlags="scroll|exitUntilCollapsed"
    app:toolbarId="@+id/toolbar">

    <ImageView
        android:id="@+id/imageHeader"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fitsSystemWindows="true"
        app:layout_collapseMode="parallax" />

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
```

A continuación, en el onCreate, vamos a rescatar la imagen

```
public class ScrollingActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_scrolling);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, text: "Replace with your own action", Snackbar.
                        .setAction( text: "Action",  listener: null).show();
            }
        });

        ImageView imageView = (ImageView) findViewById(R.id.imageHeader);

    }
```

Ahora setearemos esta imagen con la librería Glide con una imagen de Internet

https://bumptech.github.io/glide/doc/download-setup.html

Para ello:

1. Copiamos las dos líneas de instalación de la librería Glide para gestionar imágenes. Las copiamos y las pegamos en build.grade(Module.app) en dependecies, y sincronizamos.

```
repositories {
  mavenCentral()
  maven { url 'https://maven.google.com' }
}

dependencies {
  compile 'com.github.bumptech.glide:glide:4.3.1'
  annotationProcessor 'com.github.bumptech.glide:compiler:4.3.1'
}
```

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:26.+'
    compile 'com.android.support:design:26.+'
    testCompile 'junit:junit:4.12'
    compile 'com.github.bumptech.glide:glide:4.3.1'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.3.1'
}
```

2. Ahora cargamos con Glide en el onCreate cargando la imagen y una URL donde tengamos la imagen.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_scrolling);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
    fab.setOnClickListener((view) → {
            Snackbar.make(view, text: "Replace with your own action", Snackbar.LENGTH_LONG)
                    .setAction( text: "Action",  listener: null).show();
    });

    ImageView iv = (ImageView) findViewById(R.id.imageView);
    Glide.with( activity: this).load( model: "https://claxon.org/wp-content/uploads/2017/08/montar-propio-taller.jpg").into(iv);

    setTitle("Taller de 2DAM");
}
```

3. Para cambiar el texto que aparece delante de la imagen ponemos la última línea que vemos en el texto anterior.

Más opciones:

Al ejecutar vemos que el botón flotante al hacer scroll desaparece pero en verdad sigue ahí. Esto es por la opción:

```
app:layout_scrollFlags="scroll|exitUntilCollapsed"
```

Si queremos que de verás desaparezca podemos cambiarlo por la opción:

```
app:layout_scrollFlags="scroll|enterAlwaysCollapsed"
```