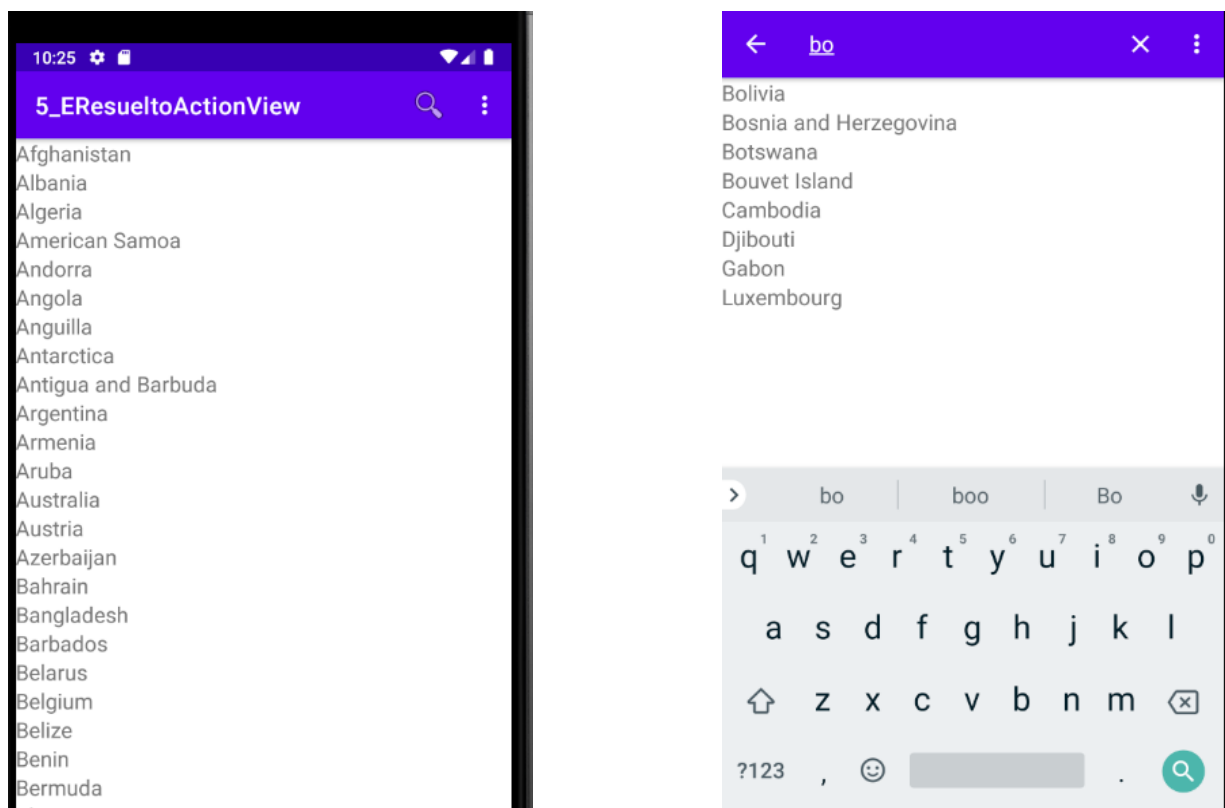


# Bloque 8 . Ejercicio Resuelto ActionView

[Descargar este ejercicio](#)

## Definición de un ActionView.

Un ActionView es un widget que se coloca en el ActionBar como sustituto de un botón de acción. Por ejemplo, si tenemos o queremos incorporar una opción de búsqueda en la ActionBar, similar a las implementadas en las guías de diseño de Android, podríamos incluir un widget SearchView en la misma, tal y como se muestra en la siguiente figura:



En nuestro caso vamos a implementar la búsqueda sobre una lista incluida en la pantalla principal, que permitirá visualizar los nombres de países que incluyan la cadena de búsqueda.

## menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:tools="http://schemas.android.com/tools"
      xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
  <item android:title="Search"
        android:id="@+id/menusearch"
        android:background="@android:color/transparent"
        android:icon="@android:drawable/ic_menu_search"
        app:showAsAction="ifRoom|collapseActionView"
        app:actionViewClass="androidx.appcompat.widget.SearchView" />
</menu>
```

El primer ítem del menú es la definición de **SearchView**, es importante resaltar que por defecto aparecerá el icono especificado en el atributo `android:icon`, y que cuando se pulsa sobre él aparecerá el widget asociado a la búsqueda, y que viene especificado en **`android:actionViewClass`**.

Para definir las acciones específicas de nuestro widget lo hacemos en la llamada al **`onCreateOptionsMenu()`**. A través del siguiente código:

```
override fun onCreateOptionsMenu(menu: Menu): Boolean {
    menuInflater.inflate(R.menu.menu, menu)
    val searchItem = menu.findItem(R.id.menusearch).actionView as SearchView
    searchItem.setOnQueryTextListener(this)
    return super.onCreateOptionsMenu(menu)
}
```

Es importante decir que nuestra clase principal implementará también la interfaz **`SearchView.OnQueryTextListener`**, y concretamente los métodos **`onQueryTextChanged()`** y **`onQueryTextSubmit()`**, que permitirán detectar las entradas en el EditText del SearchView o la salida del mismo. Será aquí donde implementaremos la funcionalidad de nuestra búsqueda en la lista de nuestra activity.

```

override fun onQueryTextChange(newText: String): Boolean {
    if (TextUtils.isEmpty(newText)) {
        adaptador = Adaptador(datos)
        recyclerView.setAdapter(adaptador)
    } else {
        val listaFiltrada = ArrayList<String>()
        for (x in datos!!) {
            val text = x.lowercase()
            ////Filtrado por contenido
            if (text.contains(newText.lowercase())) listaFiltrada.add(x)
            ////Filtadro por inicio
            else if (text.indexOf(newText.lowercase()) == 0) listaFiltrada.add(x)
            adaptador = Adaptador(listaFiltrada)
            recyclerView.setAdapter(adaptador)
        }
    }
    return false
}

```

⚠ La lista de paises se pasa como recurso