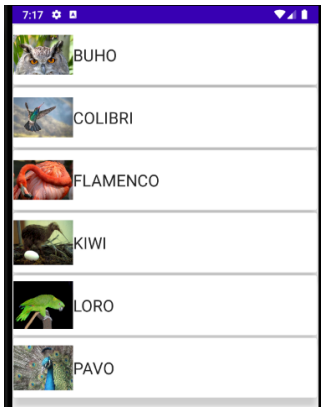
Bloc 8. Exercici Resolt "Pajaros"

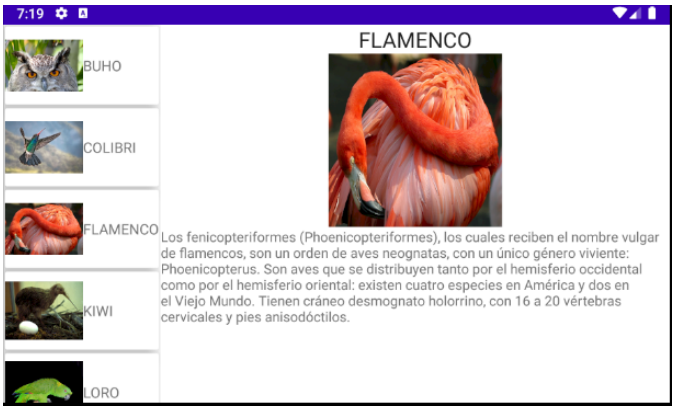
Descarregar aquest exercici

Crearem un exemple pràctic d'una llista en la qual, en seleccionar un element es mostra el detall d'aquest. L'activitat principal MainActivity, usarà diferents layouts per a les maneres portrait i landscape. Les imatges i les dades es passen com a recurs, també la classe **Dades**.

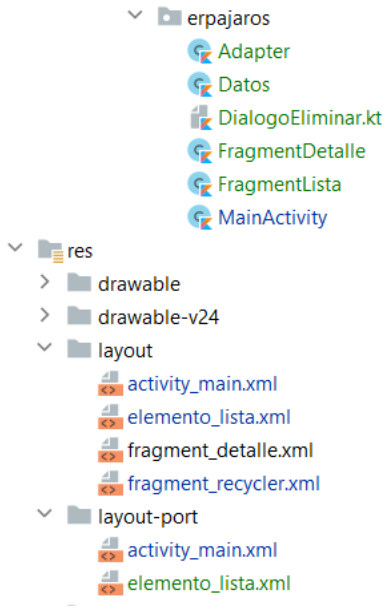
- **Portrair**, MainActivity mostrarà un Fragment amb un recycler i en seleccionar un element d'aquest, es carregarà un altre Fragment que mostrarà el detall de l'element seleccionat, com es veu en les imatges:



- **Landscape**, La activity principal carregarà dues Fragments, el de la llista i el de detall. Mostrant el detall de l'element seleccionat de la llista.



Començarem fent una descripció de les classes i arxius xml que anirem desenvolupant.



Anem primer a construir el codi de les vistes necessàries per als dos fragments:

fragment_detalle.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="center|top"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/textView_superior"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Large Text"
        android:textAppearance="?android:attr/textAppearanceLarge" />
    <ImageView
        android:id="@+id/imageView_imagen"
        android:layout_width="178dp"
        android:layout_height="178dp"
        android:scaleType="fitXY"
        android:src="@android:drawable/ic_menu_gallery" />
    <TextView
        android:id="@+id/textView_inferior"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Small Text"
        android:textAppearance="?android:attr/textAppearanceSmall" />
</LinearLayout>
```

fragment_recycler.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.recyclerview.widget.RecyclerView xmlns:android="http://schemas.android.com/apk/res/
    android:orientation="vertical" android:layout_width="match_parent"
    android:id="@+id/recycler"
    android:layout_height="match_parent">
</androidx.recyclerview.widget.RecyclerView>
```

elemento_lista.xml (port)

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="2dp"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardElevation="10dp"
    app:cardCornerRadius="2dp"
    android:id="@+id/cardView">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="left|center"
        android:orientation="horizontal">
    <ImageView
        android:id="@+id/imageView_imagen_miniatura"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@android:drawable/ic_menu_gallery" />
    <TextView
        android:id="@+id/textView_titulo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Large Text"
        android:textSize="15dp" />
    </LinearLayout>
</androidx.cardview.widget.CardView>
```

Com es pot veure en la imatge de l'estructura de les carpetes, ens trobem amb **dues activity_main.xml** i dos *element_llesta.xml", això és degut a la diferenciació que li volem donar si es canvia de vista horitzontal a vertical. Per aquest motiu haurem de crear una activity_main amb un contendre per al fragment (en aquest únic contenidor s'intercanviaran tots dos fragments) i una altra amb dos (llesta i detall).

elemento_lista.xml (land)

En aquest cas només es diferencia per la grandària del text, encara que podrien haver-hi més elements diferents.

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="2dp"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardElevation="10dp"
    app:cardCornerRadius="2dp"
    android:id="@+id/cardView">
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="left|center"
        android:orientation="horizontal">
    <ImageView
        android:id="@+id/imageView_imagen_miniatura"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@android:drawable/ic_menu_gallery" />
    <TextView
        android:id="@+id/textView_titulo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Large Text"
        android:textAppearance="?android:attr/textAppearanceLarge" />
    </LinearLayout>
</androidx.cardview.widget.CardView>
```

main_activity.xml (port)

Amb un sol contenidor per als fragments

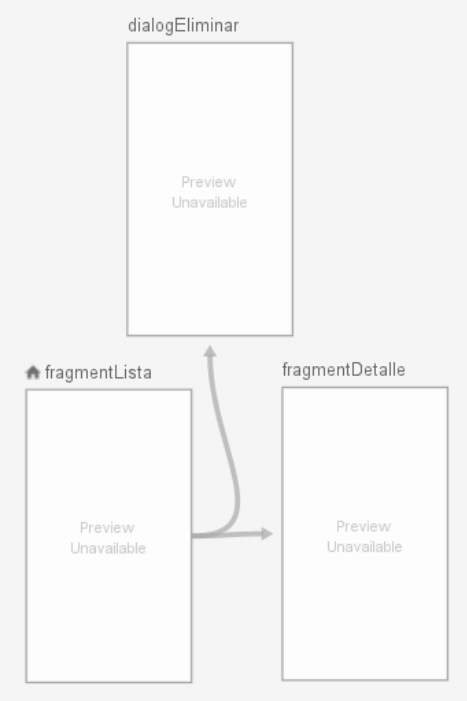
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="horizontal"
    tools:context=".MainActivity">
    <androidx.fragment.app.FragmentContainerView
        android:layout_weight="0.3"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:navGraph="@navigation/navigation"
        android:id="@+id/contenedor"/>
</LinearLayout>
```

main_activity.xml (land)

Amb un contenidor estàtic per al detall i un altre de navegació per a la llista.

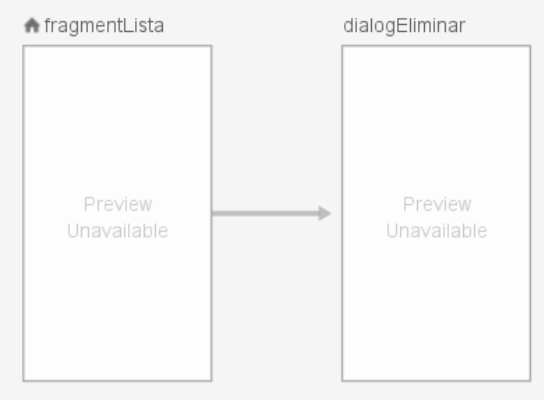
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="horizontal"
    tools:context=".MainActivity">

    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/lista"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:navGraph="@navigation/navigation_land_lista" />

    <androidx.fragment.app.FragmentContainerView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:name="com.ejemplos.b8.erpajaros.FragmentDetalle"
        android:id="@+id/detalle"/>
</LinearLayout>
```

Per a aquest exercici usarem dos elements `navigation`. Per al cas de vista **vertical** utilitzarem un gráfo de navegació com el següent:



Com es pot veure en la imatge, també haurem de crear un `FragmentDialog`, perquè es mostre un dialogue en realitzar una pulsació llarga sobre algun element de la llista per a ser eliminat.

I per a la vista horitzontal usarem un altre, només amb la part de navegació entre fragmentLista i el dialogEliminar. La navegació de l'element Detall, s'ha realitzat usant el `FragmentManager` y `FragmentTransaction`



Per a poder fer els gràfics de navegació, abans hem de crear el codi `Kotlin` necessari. Passarem a això, juntament amb els seus comentaris per línies.

Quant al codi de l'adaptador i del holder del recycler, serà idèntic de com ho fem normalment, en el qual implementem els listeners per al Click curt i llarg:

Adapter.kt

```
class Adapter(var datos: ArrayList<Datos>) :
                                RecyclerView.Adapter<Adapter.Holder>(),
                                View.OnClickListener, OnLongClickListener
{
    var listener: View.OnClickListener? = null
    var listenerLong: OnLongClickListener? = null
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
                                Holder {
        val view: View
        view = LayoutInflater.from(parent.context)
                        .inflate(R.layout.elemento_lista, parent, false)
        val holder = Holder(view)
        view.setOnClickListener(this)
        view.setOnLongClickListener(this)
        return holder
    }
    override fun onBindViewHolder(holder: Holder, position: Int) {
        (holder as Holder?)!!.bind(datos[position])
    }
    override fun getItemCount(): Int {
        return datos.size
    }
    fun miOnClick(listener: View.OnClickListener?) {
        this.listener = listener
    }
    override fun onClick(v: View) {
        if (listener != null) listener!!.onClick(v)
    }
    fun miOnLongClick(listener: OnLongClickListener?) {
        listenerLong = listener
    }
    override fun onLongClick(v: View): Boolean {
        if (listenerLong != null) listenerLong!!.onLongClick(v)
        return false
    }

    class Holder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        var lineTexto: TextView
        var imagen: ImageView
        var cardView: CardView
        fun bind(dato: Datos) {
            lineTexto.text = dato.nombre
            imagen.setImageResource(dato.icono)
        }
        init {
            lineTexto = itemView.findViewById(R.id.textView_titulo)
            imagen = itemView.findViewById(R.id.imageView_imagen_miniatura)
            cardView = itemView.findViewById(R.id.cardView)
        }
    }
}
```

La classe que implementa el detall en prémer sobre un element de la llista, quedaria amb el següent codi:

FragmentDetalle.kt

```
class FragmentDetalle : Fragment() {
    var mItem: Int? = null
    override fun onCreate(savedInstanceState: Bundle?) {

        super.onCreate(savedInstanceState)
6       if(savedInstanceState!=null) mItem=savedInstanceState?.getInt("POSICION")
        else {
            if (arguments != null) {
9               mItem = arguments?.getInt("POSICION")
10          } else mItem = 0
        }
    }
     override  fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        val rootView: View = inflater.inflate(R.layout.fragment_detalle,
                                              container, false)
        //Mostramos el contenido al usuario
        if(MainActivity.datos!!.size > 0) {
            val elemento = MainActivity.datos?.get(mItem!!)
            (rootView.findViewById<TextView>(R.id.textView_superior)).text =
                                                    elemento?.nombre
            (rootView.findViewById<TextView>(R.id.textView_inferior)).text =
                                                    elemento?.desc
            (rootView.findViewById<ImageView>(R.id.imageView_imagen)).
                                          setImageResource(elemento!!.icono)
        }
        return rootView
    }
32  override fun onSaveInstanceState(outState: Bundle) {
        super.onSaveInstanceState(outState)
        outState.putInt("POSICION", mItem!!)
    }
}
```

📌 les línies a destacar d'aquest codi són la creades per a guardar la posició premuda en determinat moment, per al cas que ocórrega algun gir de pantalla no perdre l'element premut. Guardarem la posició mitjançant el mètode `onSaveInstanceState` pel qual es passarà just abans que l'aplicació es destruïsca, **Línia 32**. Aquesta posició es podrà recuperar mitjançant el Bundle que entra en el mètode `onCreate` **Línia 6**. Les **Línies 9 i 10** són per a recuperar la

> posició quan es prem l'element de la llista en el `FragmentLista`, es recupera mitjançant la propietat `arguments`

El codi del fragment que contindrà el recycler, serà una classe que derivarà de Fragment i en la qual unflarem el layout recycler i implementarem el OnClick, manant la posició seleccionada mitjançant el Bundle al Fragment Detall:

FragmentLista.kt

```
class FragmentLista : Fragment() {
    var listenerLargo: View.OnLongClickListener? = null
    var recyclerView: RecyclerView? = null
    var valores: ArrayList<Datos>? = null
    var posicion:Int=0

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        super.onCreateView(inflater, container, savedInstanceState)
        val rootView: View = inflater.inflate(R.layout.fragment_recycler,
                                              container, false)
        recyclerView = rootView.findViewById(R.id.recycler)
        adapter = Adapter(MainActivity.datos!!)
        recyclerView?.adapter = adapter
        recyclerView?.setHasFixedSize(true)
        recyclerView?.layoutManager =LinearLayoutManager(getActivity(),
                                        LinearLayoutManager.VERTICAL, false)

        adapter!!.miOnClick { v ->
            val bundle = Bundle()
            bundle.putInt("POSICION", recyclerView!!.getChildAdapterPosition(v))
            val navController = NavHostFragment.findNavController(this)
            if (navController.currentDestination?.id == R.id.fragmentLista) {
                if (resources.configuration.orientation ==
                    Configuration.ORIENTATION_PORTRAIT) navController.navigate(
                    R.id.action_fragmentLista_to_fragmentDetalle,
                    bundle
                )
               else
                {
                    val fT = requireActivity().supportFragmentManager.
                                                            beginTransaction()
                    fT.replace(R.id.detalle,FragmentDetalle::class.java,bundle).
                                                            commit()
                }
            }
        }
        adapter!!.miOnLongClick { v ->
            val navController = NavHostFragment.findNavController(this)
            if (navController.currentDestination?.id == R.id.fragmentLista) {
                val bundle = Bundle()
                posicion=recyclerView!!.getChildLayoutPosition(v)
                bundle.putInt("POSICION",posicion)
                if (resources.configuration.orientation ==
                                                Configuration.ORIENTATION_PORTRAIT)
                 navController.navigate(R.id.action_fragmentLista_to_dialogEliminar,
                                        bundle)
                else {
```

```
                        val navHostFragment = requireActivity().
                        supportFragmentManager.findFragmentById(R.id.lista)
                        as NavHostFragment
                        val navControllerLista = navHostFragment.navController
                        navControllerLista.navigate(R.id.
                                        action_fragmentLista_to_dialogEliminar, bundle)
58                  }
                }
                true
            }

            return rootView
        }

         override fun onAttach(activity: Context) {
            super.onAttach(activity)
            try {
                valores = MainActivity.datos
                listenerLargo = activity as View.OnLongClickListener?
            } catch (e: ClassCastException) {  }
        }

        companion object {
            var adapter: Adapter? = null
        }
    }
```

📌Des de **Línia 24 a 38** en prémer sobre un element de la llista es crearà un Bundle per a guardar la posició de l'element premut, depenent si el dispositiu està en posició horitzontal o vertical: en el primer cas es navegarà al fragment detalle utilitzant l'acció que haurem creat en el graf de navegació, en el segon utilitzarem la càrrega de fragment de manera tradicional. En el codi que hi ha des de la **Línia 43 a 58**, també s'ha de tindre en compte la posició del dispositiu, però només per a usar l'un o l'altre `NavHostFragment` , en tot cas navegarem de l'element premut al dialogue que ens permet eliminar un element de la llista.

DialogEliminar.kt

```
class DialogEliminar : DialogFragment() {
    override  fun onCreateDialog(savedInstanceState: Bundle?): Dialog {
        val builder: AlertDialog.Builder = AlertDialog.Builder(getActivity())
        if (arguments != null) {
            val posicion = arguments?.getInt("POSICION")
            builder.setMessage("Deseas Eliminar el item?")
                .setTitle("AVISO")
                .setPositiveButton("OK", DialogInterface.OnClickListener {
                     dialog, id -> MainActivity.datos?.removeAt(posicion!!)
                    FragmentLista.adapter!!.notifyItemRemoved(posicion!!)})
        }
        return builder.create()
    }
}
```

> 📌Est dialogue fragment és igual als que hem vist en exercicis anteriors, passarem la posició de l'element a eliminar mitjançant un bundle

Quant al codi de la MainActivity, només s'encarregarà d'emplenar les dades passades com a recurs i de guardar una instància d'aquestes dades en cas que l'aplicació pare de manera brusca, perquè puga ser recuperada en tornar a iniciar.

```
class MainActivity : FragmentActivity(){
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        //Recuperar datos después giro
        if (savedInstanceState != null) {
            datos = savedInstanceState.getParcelableArrayList("DATOS")
        }
        else  datos = rellenarDatos()
    }

    override fun onSaveInstanceState(outState: Bundle) {
        super.onSaveInstanceState(outState)
        outState.putParcelableArrayList("DATOS", datos)
    }
    fun rellenarDatos(): ArrayList<Datos> {
        ...}

    companion object {
         var datos: ArrayList<Datos>? = null
    }
}
```