



Instituto Politécnico Nacional

Omar Montoya Romero

7CM1

Desarrollo de Aplicaciones Móviles Nativas

Sistemas Computacionales

Ejercicio6.- RecyclerView II

Mtro. Efraín Arredondo Morales

18/10/2023

Para el desarrollo del listView reutilizamos la data class para el recycler donde se define los parámetros a usar, en este caso seria el nombre y teléfono.

```
package com.example.app_09

data class Contacto(val nombre: String, val telefono: String)
```

En la parte del adaptador usamos uno diferente al de recycler, el cual tiene el mismo funcionamiento, pero es más informal, es decir la manera de pedir las cosas no son tan directas como en el recycler. Funciona de la manera de por medio del id de las variables es les manda el dato que se encuentra en la lista de contacto, dicha lista se muestra en el layout al que se le hace referencia.

```
package com.example.app_09

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.BaseAdapter
import android.widget.TextView

class ContactoAdapterListView(private val contactos: List<Contacto>) :
BaseAdapter() {

    override fun getView(position: Int, convertView: View?, parent: ViewGroup?):
View {
        val view = convertView ?:
LayoutInflater.from(parent?.context).inflate(R.layout.whatsapp_listview, parent,
false)

        val nombreContacto: TextView = view.findViewById(R.id.nombreContacto)
        val telefonoContacto: TextView = view.findViewById(R.id.telefonoContacto)

        val contacto = contactos[position]
        nombreContacto.text = contacto.nombre
        telefonoContacto.text = contacto.telefono

        return view
    }

    override fun getItem(position: Int): Any {
        return contactos[position]
    }

    override fun getItemId(position: Int): Long {
        return position.toLong()
    }

    override fun getCount(): Int {
        return contactos.size
    }
}
```

Se desarrolla el main donde declaramos la lista, la función y hacemos referencia por id al listado al que se va a mandar. En la función inicializa la lista donde debe contener los parámetros anteriormente definidos en la data class.

```
package com.example.app_09

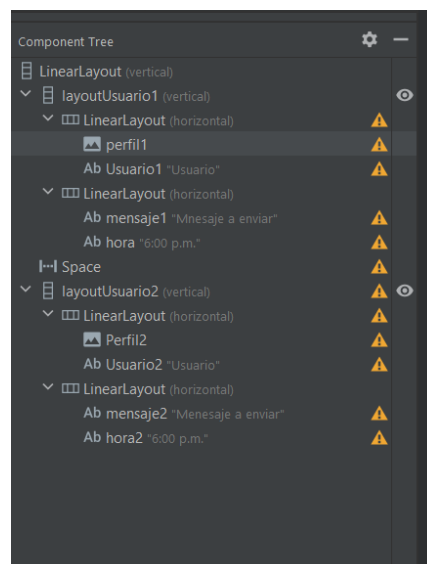
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.ListView

class ListViewActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_list_view)

        val listView: ListView = findViewById(R.id.listadoLV)
        val contactos = obtenerListaDeContactos()

        val adapter = ContactoAdapterListView(contactos)
        listView.adapter = adapter
    }
    private fun obtenerListaDeContactos(): List<Contacto> {
        return listOf(
            Contacto("Francisco Escobedo", "492-456-7890"),
            Contacto("Hector Roman", "492-654-3210"),
            Contacto("Jesus Guijarro", "492-555-1574"),
        )
    }
}
```

Para el caso del uso del recyclerView se copio y reutilizo el que se realizó en clase, mientras que para el desarrollo del chat de WhatsApp se utilizó el recyclerView pero se hicieron algunas modificaciones para el correcto funcionamiento del chat, se planteó hacerlo en ítems separados pero al final por facilidad se realizó en un solo ítem el cual cuenta con los siguientes componentes:



Para la data class se definieron 2 usuarios, 2 mensajes, 2 horas, esto para simular cada usuario con su mensaje y la hora de envío.

```
package com.example.app_09

data class Mensajes(val usuario: String, val mensaje: String, val hora:String, val usuario2:
String, val mensaje2: String, val hora2:String)
```

Para el Adapter se utilizo como base el que se desarrollo en clase con la diferencia de que se uso una propiedad de los linearlayout el cual fue si son visibles o no, esto nos permitió poner if para cuando un usuario no mando nada el mensaje no aparezca y automáticamente muestre el otro mensaje sin mostrar el fondo vacio del linearlayout. Se hace referencia al recycler a usar y a la interfaz desarrollada por id a cada componente.

```
package com.example.app_09

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.LinearLayout
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

class ChatAdapterRecycler(private val mensajes:
List<Mensajes>):RecyclerView.Adapter<ChatAdapterRecycler.ViewHolder>() {
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
ChatAdapterRecycler.ViewHolder {
        val view = LayoutInflater.from(parent.context)
            .inflate(R.layout.whatsapp_app_message, parent, false)
        return ViewHolder(view)
    }

    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        val mensaje = mensajes[position]
        if (mensaje.usuario.isNotEmpty() && mensaje.mensaje.isNotEmpty() &&
mensaje.hora.isNotEmpty()) {
            holder.layoutUsuario1.visibility = View.VISIBLE
            holder.usuario1.text = mensaje.usuario
            holder.mensaje1.text = mensaje.mensaje
            holder.hora1.text = mensaje.hora
        } else {
            holder.layoutUsuario1.visibility = View.GONE
        }

        if (mensaje.usuario2.isNotEmpty() && mensaje.mensaje2.isNotEmpty() &&
mensaje.hora2.isNotEmpty()) {
            holder.layoutUsuario2.visibility = View.VISIBLE
            holder.usuario2.text = mensaje.usuario2
            holder.mensaje2.text = mensaje.mensaje2
            holder.hora2.text = mensaje.hora2
        } else {
            holder.layoutUsuario2.visibility = View.GONE
        }
    }

    override fun getItemCount(): Int {
        return mensajes.size
    }

    inner class ViewHolder (itemView: View): RecyclerView.ViewHolder(itemView) {
        val usuario1: TextView = itemView.findViewById(R.id.Usuario1)
        val mensaje1: TextView = itemView.findViewById(R.id.mensaje1)
        val hora1: TextView = itemView.findViewById(R.id.hora)
        val layoutUsuario1: LinearLayout = itemView.findViewById(R.id.layoutUsuario1)
        val layoutUsuario2: LinearLayout = itemView.findViewById(R.id.layoutUsuario2)
        val usuario2: TextView = itemView.findViewById(R.id.Usuario2)
        val mensaje2: TextView = itemView.findViewById(R.id.mensaje2)
```

```

        val hora2: TextView = itemView.findViewById(R.id.hora2)
    }
}

```

En el Activity del WhatsApp se declaran el textview donde se ingresa el mensaje, se declara el botón de enviar esto para darle un poco de funcionalidad a esa parte, independientemente de lo que se mande no se va a mostrar en el chat ya que eso no está configurado.

```

package com.example.app_09

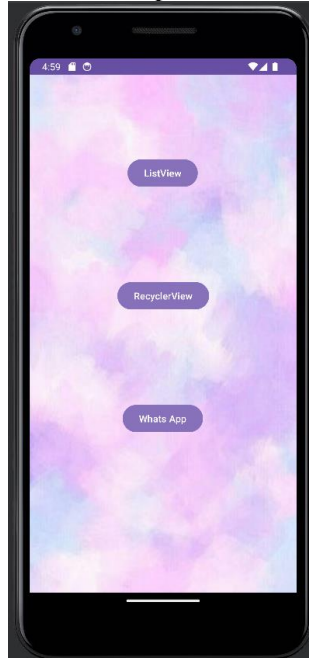
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView

class WhatsAppActivity : AppCompatActivity() {
    lateinit var btnEnviar:Button
    lateinit var tvMensaje:TextView
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_whats_app)
        btnEnviar = findViewById(R.id.btnEnviar)
        tvMensaje = findViewById(R.id.tvMensajes)
        // Crear una lista de contactos ficticios
        val mensajes = listOf(
            Mensajes("Francisco Escobedo", "492-456-7890", "12:00", "Francisco Escobedo",
"492-456-7890", "12:00"),
            Mensajes("", "", "", "Francisco Escobedo", "492-456-7890", "12:00"),
            Mensajes("Francisco Escobedo", "492-456-7890", "12:00", "", "", ""),
            Mensajes("Francisco Escobedo", "492-456-7890", "12:00", "", "", ""),
            Mensajes("Francisco Escobedo", "492-456-7890", "12:00", "Francisco Escobedo",
"492-456-7890", "12:00"),
            Mensajes("Francisco Escobedo", "492-456-7890", "12:00", "Francisco Escobedo",
"492-456-7890", "12:00"),
            Mensajes("Francisco Escobedo", "492-456-7890", "12:00", "Francisco Escobedo",
"492-456-7890", "12:00"),
        )
        // Obtener una referencia al RecyclerView
        val recyclerView = findViewById<RecyclerView>(R.id.recyclerWhatsApp)

        // Crear y configurar el adaptador
        val adapter = ChatAdapterRecycler(mensajes)
        recyclerView.adapter = adapter
        recyclerView.layoutManager = LinearLayoutManager(this)
        btnEnviar.setOnClickListener {
            tvMensaje.text = ""
        }
    }
}

```

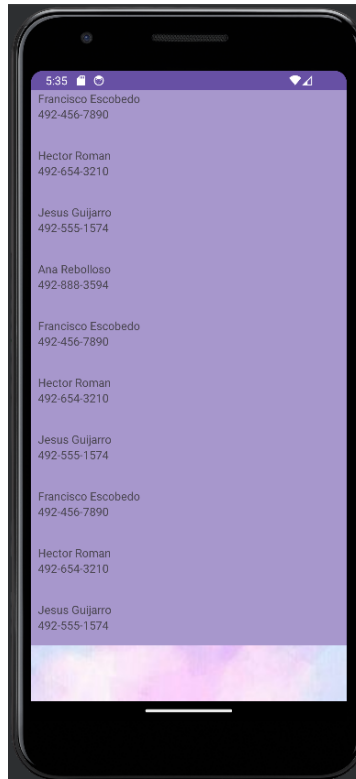
En la primer imagen se muestra el menú para acceder a los diferentes componentes



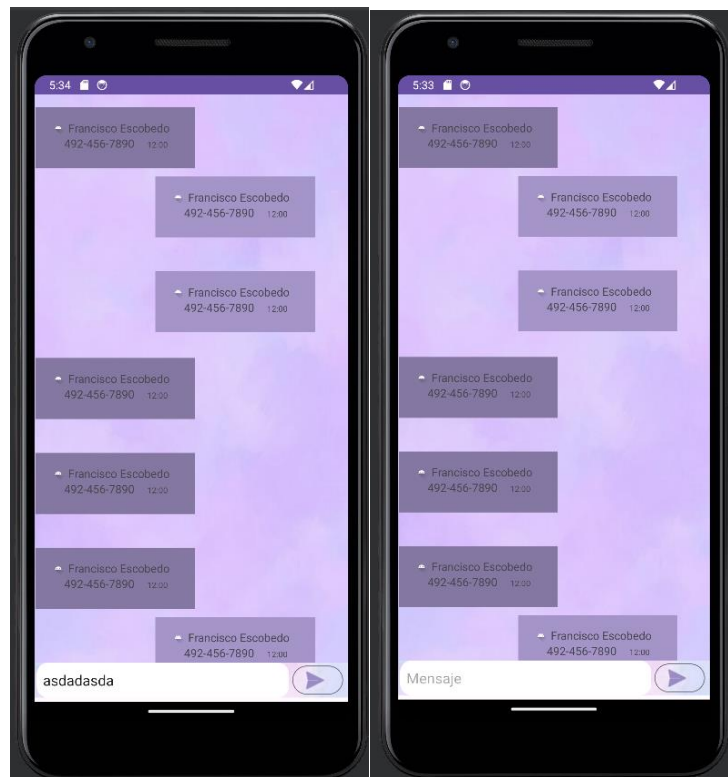
En la segunda imagen se muestra el listView el cual cuando se le intenta dar scroll no hace ninguna animación, se le pusieron menos contenido para evitar confundirlo con el recycler ya que en vista son muy parecidos.



En la tercer imagen se muestra el recyclerView el cual tiene mas contactos para diferenciarlo más fácil, al intentar hacer scroll este tiene una pequeña animación donde intenta hacer mas scroll aunque no se encuentren mas componentes a mostrar, lo contrario al listView.



En la imágenes restantes se muestra el WhatsApp donde se muestra el funcionamiento del adapter al mandar mensajes un solo usuario o al mandarlo los 2. Así mismo muestra el funcionamiento del botón de enviar.



El color del diseño se decidió al ser el color que se le asignó automáticamente al icono para que combinara se acomodó la paleta de colores para que combinara y no se viera raro, así mismo el diseño de los linearlayout se decidieron de la forma que tienen en el WhatsApp original.