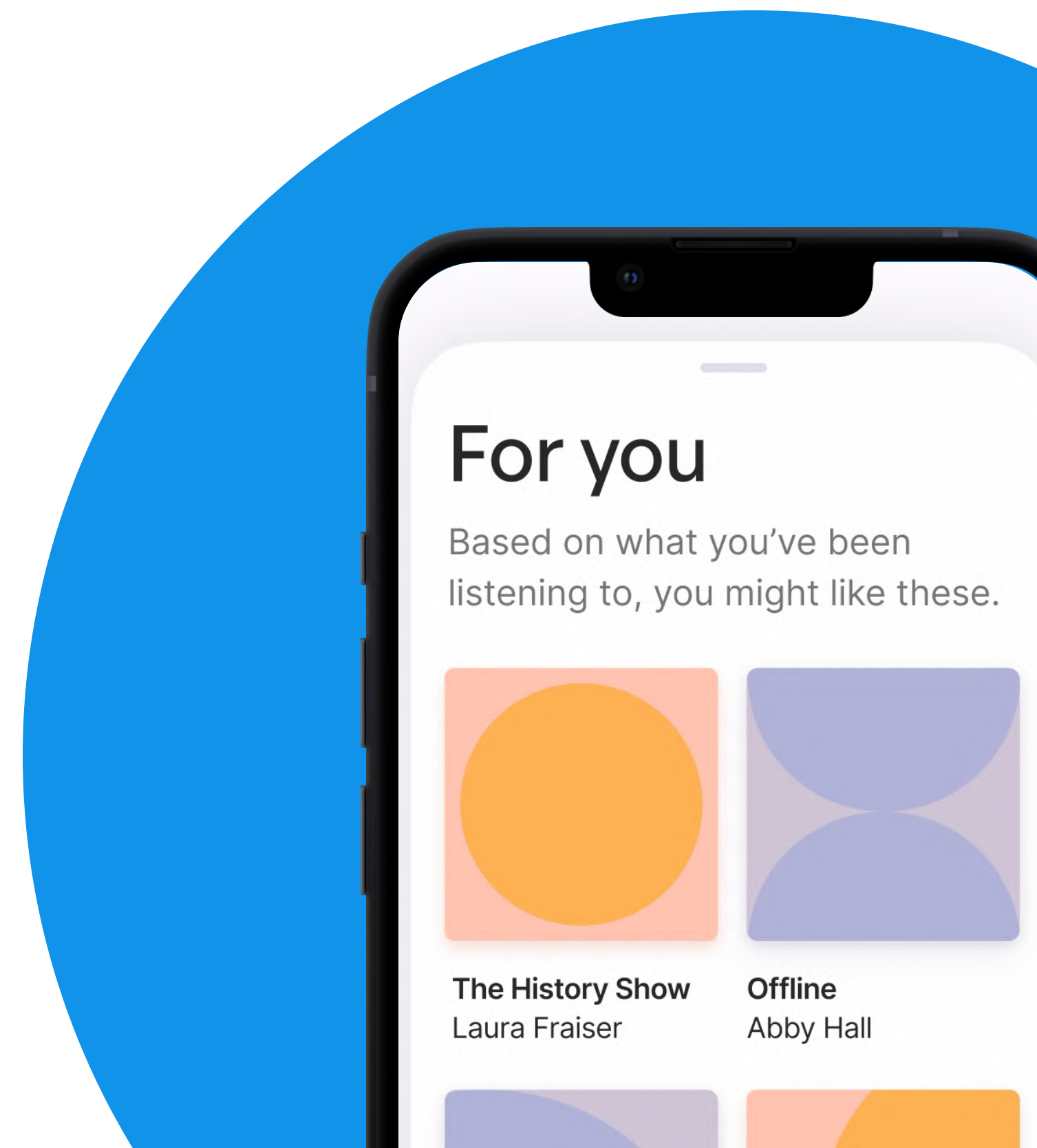


ANDROID EN KOTLIN

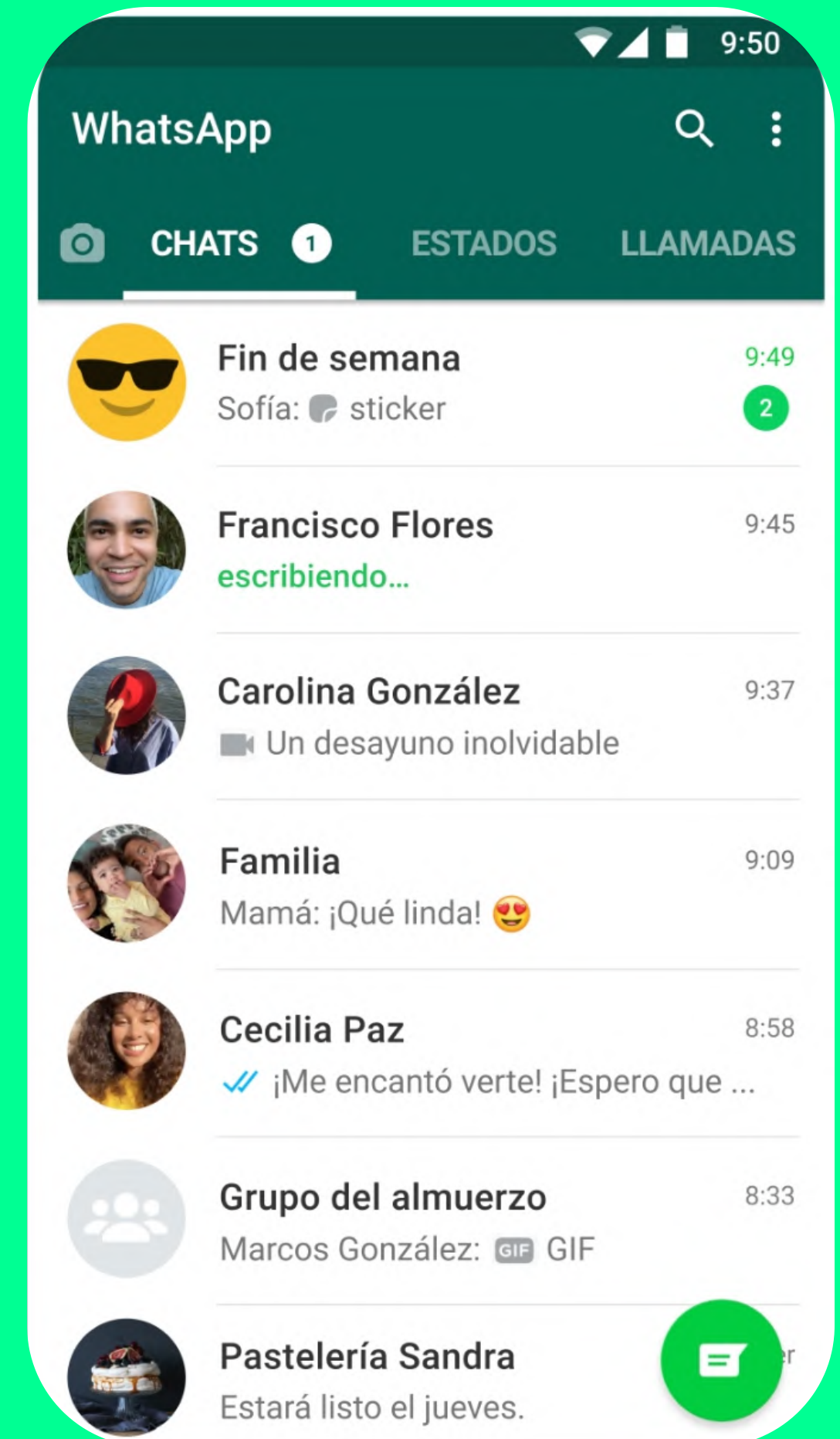
RECYCLER VIEW



RECYCLERVIEW

El RecyclerView es un simple listado que puede estar en vertical, horizontal y en cuadrícula, pocas apps no tienen este componente de Android. Debe su nombre de que cuando haces scroll, recicla las vistas y gasta así poca memoria.

<https://developer.android.com/guide/topics/ui/layout/recyclerview>



IMPLEMENTACIÓN 1/4

Implementamos el componente en un Activity o Fragment

```
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/mainRecyclerView"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
/>
```

IMPLEMENTACIÓN 2/4

El siguiente paso es crear un Adapter para gestionar nuestro RecyclerView y un ViewHolder encargado de asignar los valores visuales.

```
class MainAdapter(private val mDataSet: List<String>) :
    RecyclerView.Adapter<MainAdapter.MainViewHolder>() {

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MainViewHolder {
        val v = LayoutInflater.from(parent.context).inflate(R.layout.item_block, parent, false)
        return MainViewHolder(v)
    }

    override fun onBindViewHolder(holder: MainViewHolder, position: Int) {
        val data = mDataSet.get(position)
        holder.bindItems(data)
    }

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

    inner class MainViewHolder(var v: View) : RecyclerView.ViewHolder(v) {
        val txt = v.findViewById<TextView>(R.id.mytext)

        fun bindItems(data: String) {
            txt.text = data
        }
    }
}
```


IMPLEMENTACIÓN 3/4

Nuestro Adapter necesita que le asignemos un componente visual a repetir, para ello tendríamos que crear el layout que se repetirá en nuestro listado. En mi caso lo he llamado item_block.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_margin="5dp"
    android:background="#293379"
    android:layout_height="wrap_content">
    <TextView
        android:id="@+id/mytexto"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:text="Hola"
        android:textColor="@android:color/white"
        android:textSize="16sp" />
</LinearLayout>
```

IMPLEMENTACIÓN 4/4

Por último, instanciamos el adapter nada más iniciarse la app, justamente en el onCreate. En este caso tenemos una cuadrícula de 2.

```
class MainActivity : AppCompatActivity() {  
  
    var demoNames = listOf(  
        "Jamey Bush", "Casandra Red", "Melvin Detrick", "Mirella Jiggetts", "Brook Hetzel",  
        "Eva Mccrystal", "Glennie Hiott", "Alverta Ruggles", "Floria Pedroza", "Marianela Redr",  
        "Colby Bellew", "Marquerite Kite", "Marcelene Rhoads", "Taneka Burgin",  
        "Marci Smits", "Michelle Madero", "Pinkie Josey", "Marlys Nieman", "Ling Reddick"  
    )  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        val mainRecyclerView = findViewById<RecyclerView>(R.id.mainRecyclerView)  
  
        val mAdapter = MainAdapter(demoNames)  
        mainRecyclerView.layoutManager = GridLayoutManager(this, 2)  
        mainRecyclerView.adapter = mAdapter  
    }  
}
```

AVANZAMOS

TAMBIÉN TENEMOS DOS OPCIONES MÁS

LISTADO VERTICAL Y HORIZONTAL

En este código, poniendo "VERTICAL" u "HORIZONTAL" podrás ver que cambia la forma de mostrarse el listado.

```
class MainActivity : AppCompatActivity() {

    var demoNames = listOf(
        "Jamey Bush", "Casandra Red", "Melvin Detrick", "Mirella Jiggetts", "Brook Hetzel",
        "Eva Mccrystal", "Glennie Hiott", "Alverta Ruggles", "Floria Pedroza", "Marianela Redman",
        "Colby Bellew", "Marquerite Kite", "Marcelene Rhoads", "Taneka Burgin",
        "Marci Smits", "Michelle Madero", "Pinkie Josey", "Marlys Nieman", "Ling Reddick"
    )

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val mainRecyclerView = findViewById<RecyclerView>(R.id.mainRecyclerView)

        val mAdapter = MainAdapter(demoNames)
        mainRecyclerView.layoutManager = LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false)
        mainRecyclerView.adapter = mAdapter
    }
}
```


AVANZAMOS

COMO GESTIONAR EL CLICK

Lo que haremos es pasarle por parámetro al Adapter una función, es un medio común para que los objetos no relacionados se comuniquen entre sí.

Podemos ponerle los parámetros que queramos y las funciones que queramos.

La función no tiene lógica
solo es para transmitir parámetros entre clases, la lógica se aplicará cuando se use.

```
class MainAdapter(private val mDataSet: List<String>, var click: (String) -> Unit) :  
    RecyclerView.Adapter<MainAdapter.MainViewHolder>() {  
  
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MainViewHolder {  
        val v = LayoutInflater.from(parent.context).inflate(R.layout.item_block, parent, false)  
        return MainViewHolder(v)  
    }  
  
    override fun onBindViewHolder(holder: MainViewHolder, position: Int) {  
        val data = mDataSet.get(position)  
        holder.bindItems(data)  
    }  
  
    override fun getItemCount(): Int {  
        return mDataSet.size  
    }  
}
```

PRÁCTICA

Replicar el recyclerview de Whatsapp o Telegram (o de alguna app que te guste) en 30 minutos.

No incluyas otros elementos como tabs, utiliza solo lo que sepas y céntrate en el recyclerview.



A large, solid green circle is centered on a white background. Inside the circle, the word "GRACIAS" is written in a bold, black, sans-serif font, centered horizontally and vertically.

GRACIAS