Desarrollo Android

Clase 07

RecyclerView

Pre-RecyclerView

ListView

Es la antecesora de las RecyclerViews.

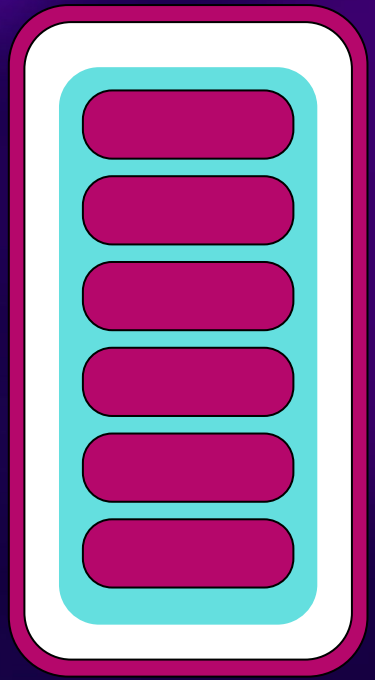
Permitía desplegar listados verticales fácilmente.

No existía forma oficial de crear listados horizontales.

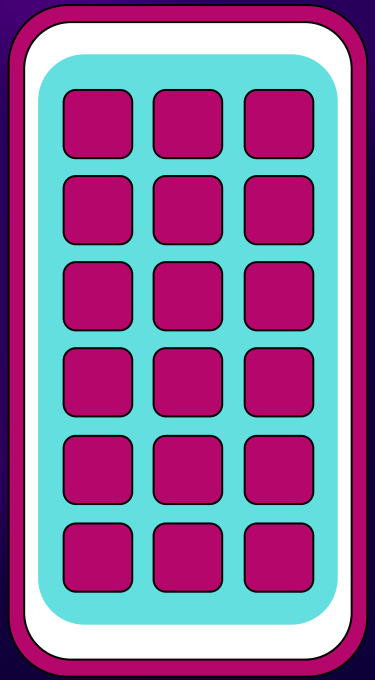
Para crear grillas era necesario usar GridViews

Por defecto más costosa en términos de performance.

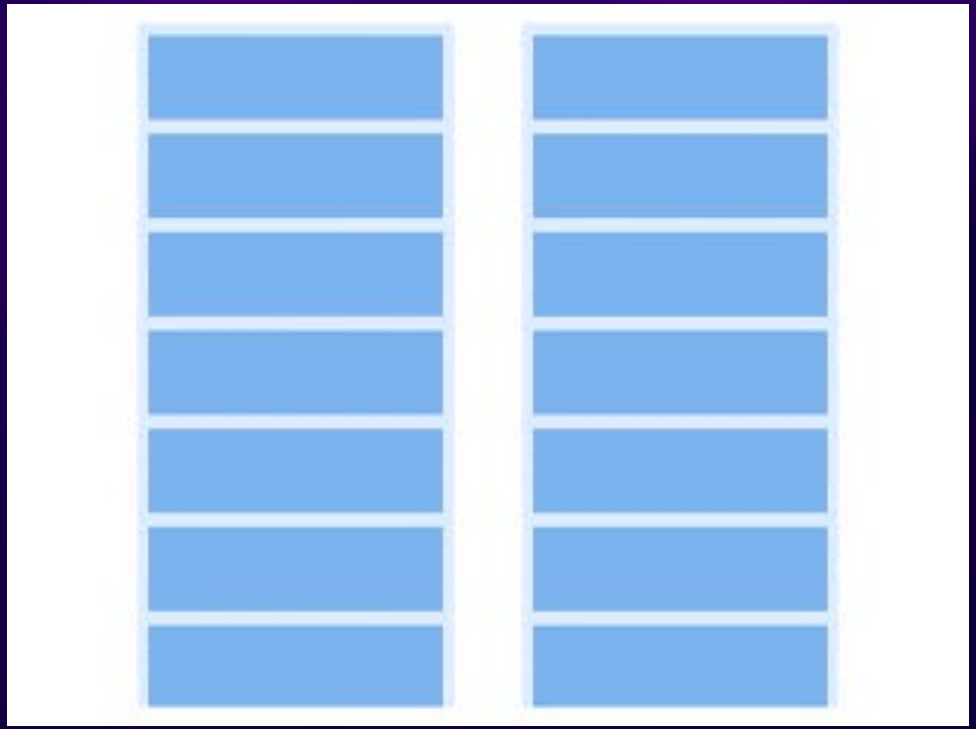
ListView



GridView



RecyclerView



Componentes

Datos

Estructura o estructuras de datos dentro de las cuales se contienen los elementos a mostrar en la RecyclerView.

Los datos pueden ser extraídos de una Base de Datos, traídos de una API o incluso creados en memoria.

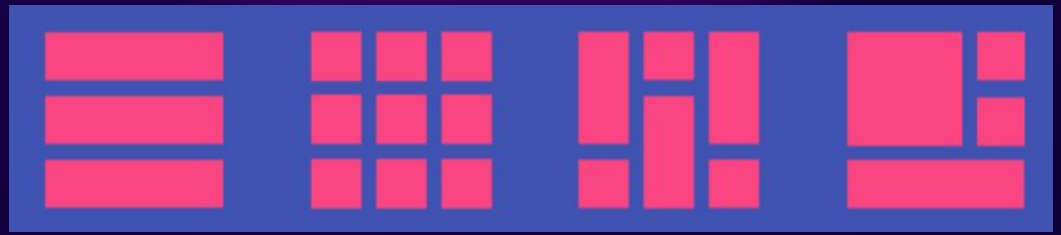
Adapter

Asocia elementos procedentes de los datos con elementos visuales de la RecyclerView

LayoutManager

Encargado de ubicar a los elementos visuales dentro de la RecyclerView.

Nos permite crear listados verticales/horizontales, grillas, y cualquier otra organización que se nos ocurra.



ViewHolder

Un ViewHolder es un componente que almacena la estructura de un elemento de la RecyclerView (View) para ser reutilizada en cada nuevo elemento desplegado, en lugar de “inflar” el layout una y otra vez.

La RecyclerView crea tantos ViewHolders como sea necesario para llenar la pantalla (y mantener una buena UX)

Item Layout

Es el layout que será utilizado para representar a cada elemento de la lista.

Cómo construimos un RecyclerView?

Agregar la RecyclerView al layout

```
<androidx.constraintlayout.widget.ConstraintLayout
...
>

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        ...
        tools:listitem="@layout/list_item" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Layout para los elementos del listado

```
<androidx.constraintlayout.widget.ConstraintLayout
...
>

    <ImageView
        android:id="@+id/imageView"
        ...
        tools:srcCompat="@tools:sample/avatars" />

    <TextView
        android:id="@+id/textView"
        ...
        tools:text="John Doe" />

    <TextView
        android:id="@+id/textView2"
        ...
        tools:text="21 years" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Adapter

```
class PeopleAdapter(private val list: List<Person>): RecyclerView.Adapter<PeopleAdapter.PeopleViewHolder>() {

    class PeopleViewHolder(itemView: View): RecyclerView.ViewHolder(itemView) {
        val nameText: TextView = itemView.findViewById(R.id.textView)
        val ageText: TextView = itemView.findViewById(R.id.textView2)
    }
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PeopleViewHolder {
        val itemView = LayoutInflater.from(parent.context).inflate(R.layout.list_item, parent, false)

        return PeopleViewHolder(itemView)
    }
    override fun onBindViewHolder(holder: PeopleViewHolder, position: Int) {
        val item = list[position]
        holder.nameText.text = item.fullName
        holder.ageText.text = "${item.age} años"
    }
    override fun getItemCount(): Int {
        return list.size
    }
}
```

Adapter

```
class PeopleAdapter(private val list: List<Person>): RecyclerView.Adapter<PeopleAdapter.PeopleViewHolder>() {

    class PeopleViewHolder(itemView: View): RecyclerView.ViewHolder(itemView) {
        val nameText: TextView = itemView.findViewById(R.id.textView)
        val ageText: TextView = itemView.findViewById(R.id.textView2)
    }
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PeopleViewHolder {
        val itemView = LayoutInflater.from(parent.context).inflate(R.layout.list_item, parent, false)

        return PeopleViewHolder(itemView)
    }
    override fun onBindViewHolder(holder: PeopleViewHolder, position: Int) {
        val item = list[position]
        holder.nameText.text = item.fullName
        holder.ageText.text = "${item.age} años"
    }
    override fun getItemCount(): Int {
        return list.size
    }
}
```

Adapter

```
class PeopleAdapter(private val list: List<Person>): RecyclerView.Adapter<PeopleAdapter.PeopleViewHolder>() {

    class PeopleViewHolder(itemView: View): RecyclerView.ViewHolder(itemView) {
        val nameText: TextView = itemView.findViewById(R.id.textView)
        val ageText: TextView = itemView.findViewById(R.id.textView2)
    }
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PeopleViewHolder {
        val itemView = LayoutInflater.from(parent.context).inflate(R.layout.list_item, parent, false)

        return PeopleViewHolder(itemView)
    }
    override fun onBindViewHolder(holder: PeopleViewHolder, position: Int) {
        val item = list[position]
        holder.nameText.text = item.fullName
        holder.ageText.text = "${item.age} años"
    }
    override fun getItemCount(): Int {
        return list.size
    }
}
```

Adapter

```
class PeopleAdapter(private val list: List<Person>): RecyclerView.Adapter<PeopleAdapter.PeopleViewHolder>() {

    class PeopleViewHolder(itemView: View): RecyclerView.ViewHolder(itemView) {
        val nameText: TextView = itemView.findViewById(R.id.textView)
        val ageText: TextView = itemView.findViewById(R.id.textView2)
    }
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PeopleViewHolder {
        val itemView = LayoutInflater.from(parent.context).inflate(R.layout.list_item, parent, false)

        return PeopleViewHolder(itemView)
    }
    override fun onBindViewHolder(holder: PeopleViewHolder, position: Int) {
        val item = list[position]
        holder.nameText.text = item.fullName
        holder.ageText.text = "${item.age} años"
    }
    override fun getItemCount(): Int {
        return list.size
    }
}
```

Adapter

```
class PeopleAdapter(private val list: List<Person>): RecyclerView.Adapter<PeopleAdapter.PeopleViewHolder>() {

    class PeopleViewHolder(itemView: View): RecyclerView.ViewHolder(itemView) {
        val nameText: TextView = itemView.findViewById(R.id.textView)
        val ageText: TextView = itemView.findViewById(R.id.textView2)
    }
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PeopleViewHolder {
        val itemView = LayoutInflater.from(parent.context).inflate(R.layout.list_item, parent, false)

        return PeopleViewHolder(itemView)
    }
    override fun onBindViewHolder(holder: PeopleViewHolder, position: Int) {
        val item = list[position]
        holder.nameText.text = item.fullName
        holder.ageText.text = "${item.age} años"
    }
    override fun getItemCount(): Int {
        return list.size
    }
}
```

Adapter

```
class PeopleAdapter(private val list: List<Person>): RecyclerView.Adapter<PeopleAdapter.PeopleViewHolder>() {

    class PeopleViewHolder(itemView: View): RecyclerView.ViewHolder(itemView) {
        val nameText: TextView = itemView.findViewById(R.id.textView)
        val ageText: TextView = itemView.findViewById(R.id.textView2)
    }
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PeopleViewHolder {
        val itemView = LayoutInflater.from(parent.context).inflate(R.layout.list_item, parent, false)

        return PeopleViewHolder(itemView)
    }
    override fun onBindViewHolder(holder: PeopleViewHolder, position: Int) {
        val item = list[position]
        holder.nameText.text = item.fullName
        holder.ageText.text = "${item.age} años"
    }
    override fun getItemCount(): Int {
        return list.size
    }
}
```

Wrapping up

```
class MainActivity : AppCompatActivity() {

    lateinit var recyclerView: RecyclerView

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

        recyclerView = findViewById(R.id.recyclerView)

        val people = listOf(...)

        recyclerView.layoutManager = LinearLayoutManager(this)
        recyclerView.adapter = PeopleAdapter(people)
    }
}
```