Aula 22

RecyclerView & Sprint 0

DigitalHouse > Coding School

Sprint 0

Datas:

- Início 19/10/2020 até 09/11/2020

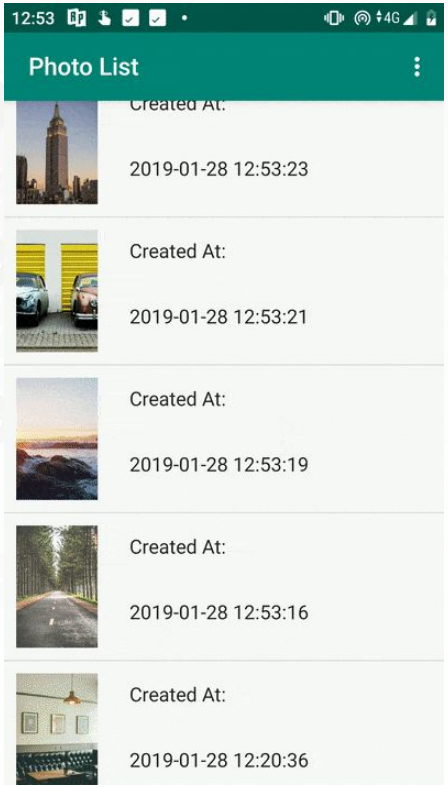
Entrega:

- Desenvolvimento dos protótipos

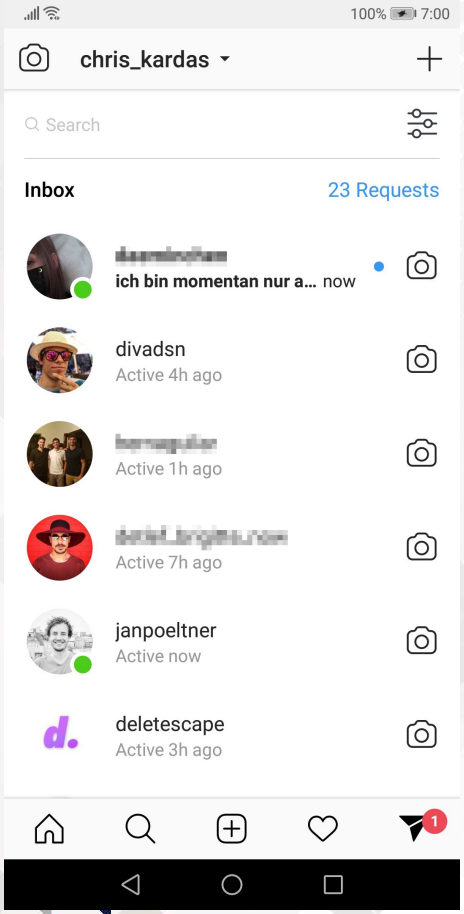
Avisos

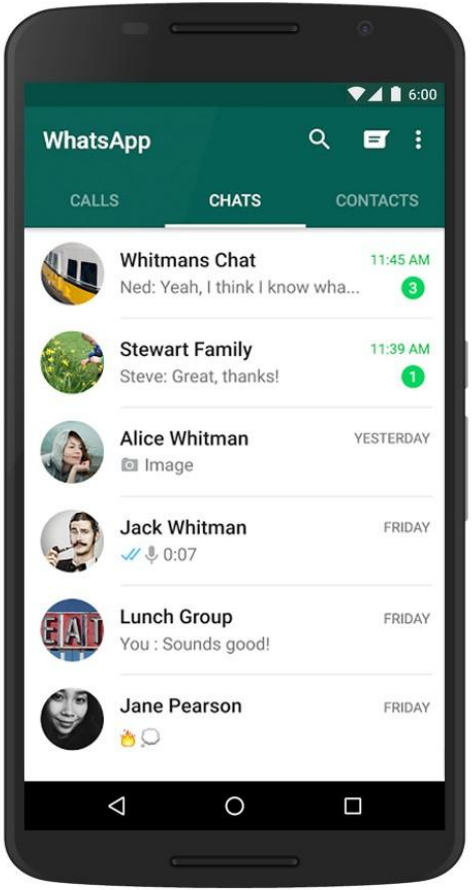
- Android Studio 4.1 (bugado)
- Mudança brusca de estrutura
- Uso excessivo de CPU
- synthetic foi depreciado pela Google

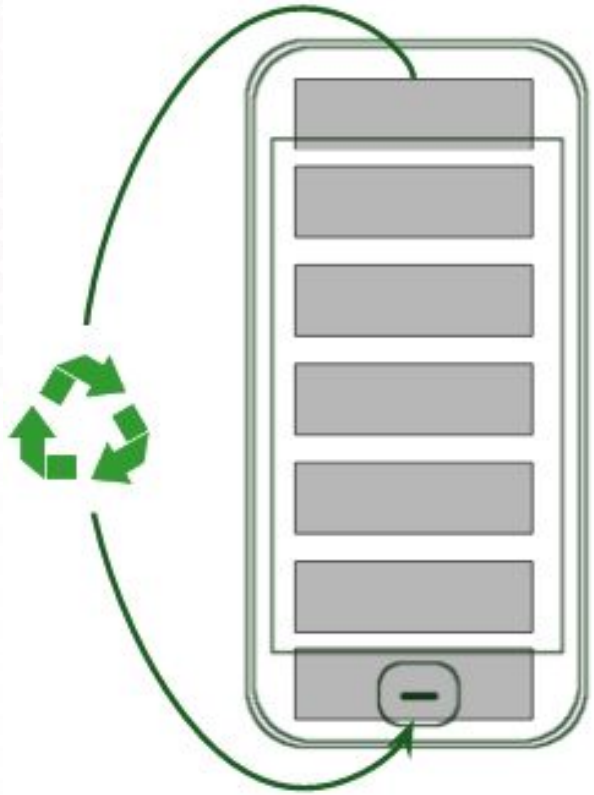
RecyclerView



DigitalHouse > Coding School



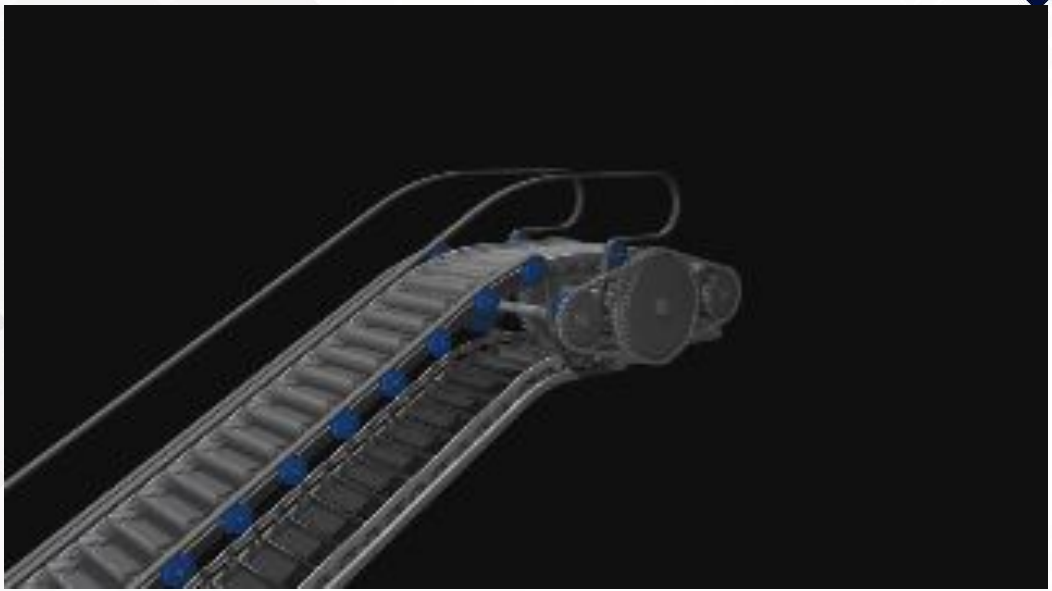




DigitalHouse >
Coding School

Reciclagem de views

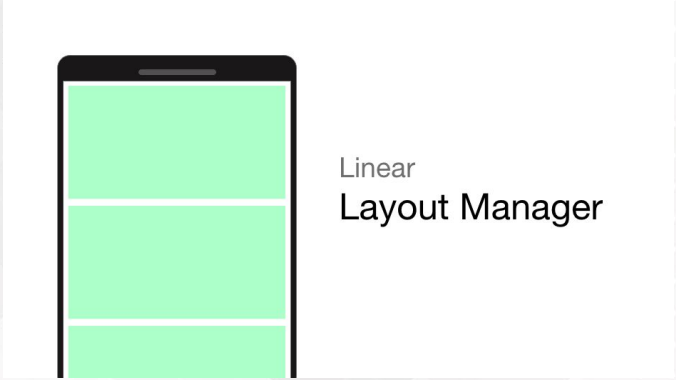
Ajuda a exibir listas longas dentro dos aplicativos re-utilizando as Views que não estão visíveis na tela naquele momento. Similar ao como uma escada rolante funciona.

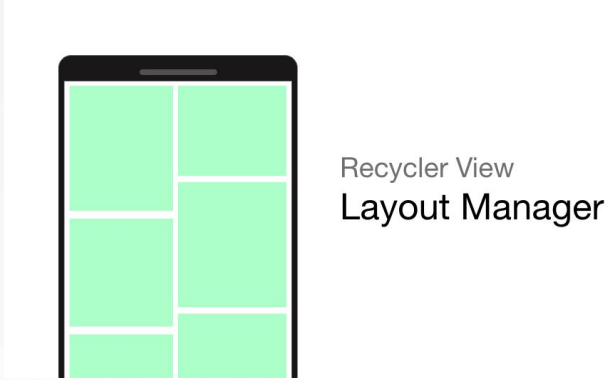


https://developer.android.com/guide/topics/ui/layout/recyclerview

Estrutura de um RecyclerView

Tipos de LayoutManager





Adapter

- É quem adapta o Recycler e os dados que serão apresentados nele, o intermediário que agrega a lógica e o layout.
- Responsável por criar uma View para cada item do conjunto de dados.

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

DataSet

- Conjunto de dados a serem utilizados

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

Dependência

Inserir dependência no build.gradle

```
implementation "androidx.recyclerview:recyclerview:1.1.0"
```

```kotlin
class ActRecycler : AppCompatActivity() {

    private val recycler by lazy {
        RecyclerView( context: this)
    }
    private val manager by lazy {
        LinearLayoutManager( context: this, VERTICAL, reverseLayout: false)
    }

    private val alunos = listOf(
        "Rafinha",
        "Henrique",
        "Xandão",
        "Daniel",
        "Peter Henry",
        "Giulia",
        "Felipe"
    )

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(recycler)

        recycler.layoutManager = manager
        recycler.adapter = AdapterAlunos( context: this, alunos)
    }
}
```

```kotlin
class AlunoViewHolder(view: View) : RecyclerView.ViewHolder(view) {
    var textView: TextView = view.findViewById(R.id.textView)
    var cardView: CardView = view.findViewById(R.id.cardView)
}
```

```xml
<androidx.recyclerview.widget.RecyclerView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/recycler"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    />
```

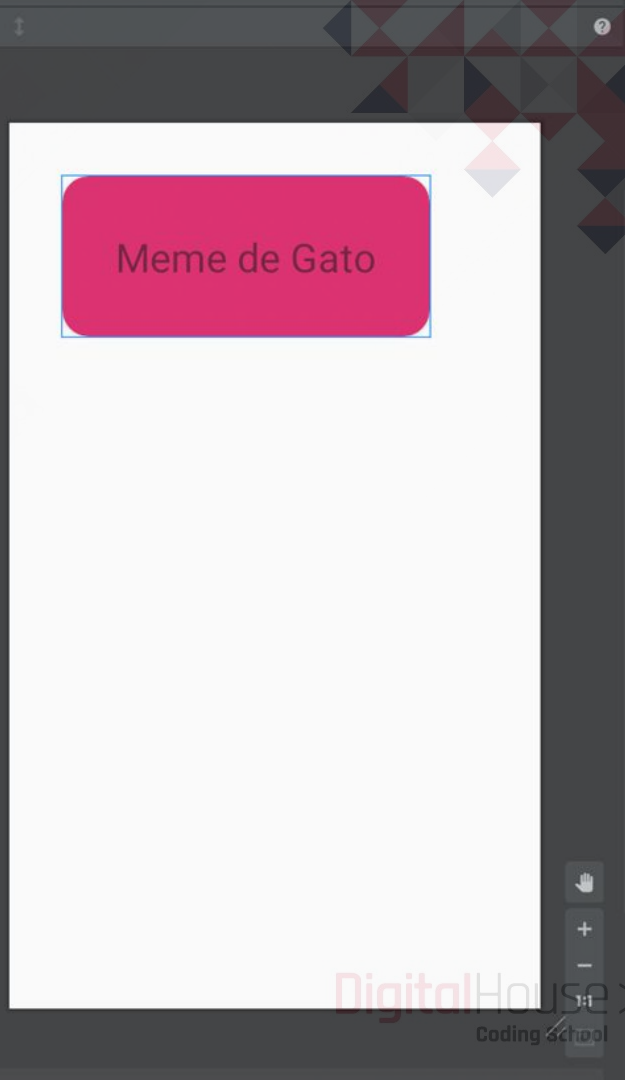
```xml
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="32dp"
    >

    <androidx.cardview.widget.CardView
        android:id="@+id/cardView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:cardBackgroundColor="@color/pink_600"
        app:cardCornerRadius="16dp"
        >

        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:gravity="center"
            android:padding="32dp"
            android:textSize="24sp"
            tools:text="Meme de Gato"
            />

    </androidx.cardview.widget.CardView>

</FrameLayout>
```



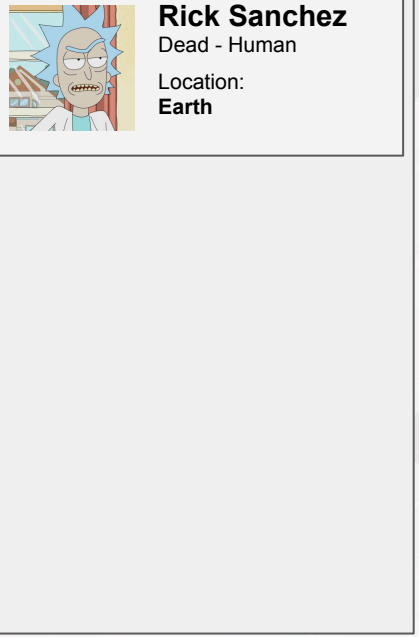
```
class AdapterAlunos(val context : Context, private val listaDeAlunos: List<String>) :
    RecyclerView.Adapter<AlunoViewHolder>() {

    override fun onCreateViewHolder(group: ViewGroup, type: Int): AlunoViewHolder {
        return AlunoViewHolder(
            LayoutInflater.from(context)
                .inflate(R.layout.texto_pequeno, group, attachToRoot: false)
        )
    }

    override fun onBindViewHolder(holder: AlunoViewHolder, position: Int) {
        val nomeDoAluno : String  = listaDeAlunos[position]
        holder.textView.text = nomeDoAluno
        holder.textView.setOnClickListener { it: View!
            context.startActivity(
                Intent(context, ActRevisaoFrags::class.java)
                    .putExtra( name: "nome", nomeDoAluno)
            )
        }
    }

    override fun getItemCount() : Int  = listaDeAlunos.size
}
```

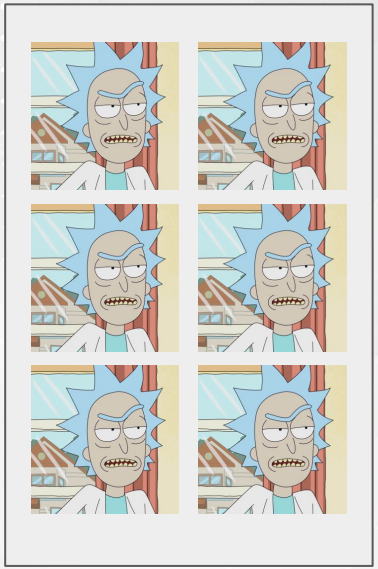
Exercício 1



Implementar uma listagem com os personagens do Rick and Morty

- Estrutura de card

Exercício 2



Implementar um grid com os personagens do Rick and Morty