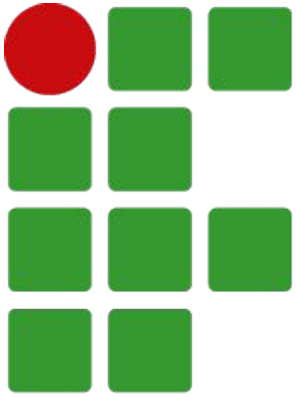


INSTITUTO FEDERAL
Paraná
Campus Paranaguá

RecyclerView

Dispositivos Móveis

Prof. Diego Stiehl

INSTITUTO FEDERAL

Listas

- **Listagens de conjuntos** de dados são importantes para nossas aplicações
  - Android ou outras
- Servem para mostrar itens relacionados de maneira padronizada entre si
  - Lista de produtos, com nome, preço e foto
  - Lista de clientes
  - Grade de jogos com imagem e nome
  - …

INSTITUTO FEDERAL

List e Grid

- O Android oferece por padrão duas opções de containers para listagem de dados
  - ListView
    - Mostra os dados um abaixo do outro
  - GridView
    - Mostra os dados como uma grade
- A partir do Android 5.0 (Lollipop)
  - **RecyclerView**
    - View otimizada que agrupa a funcionalidade dos dois componentes

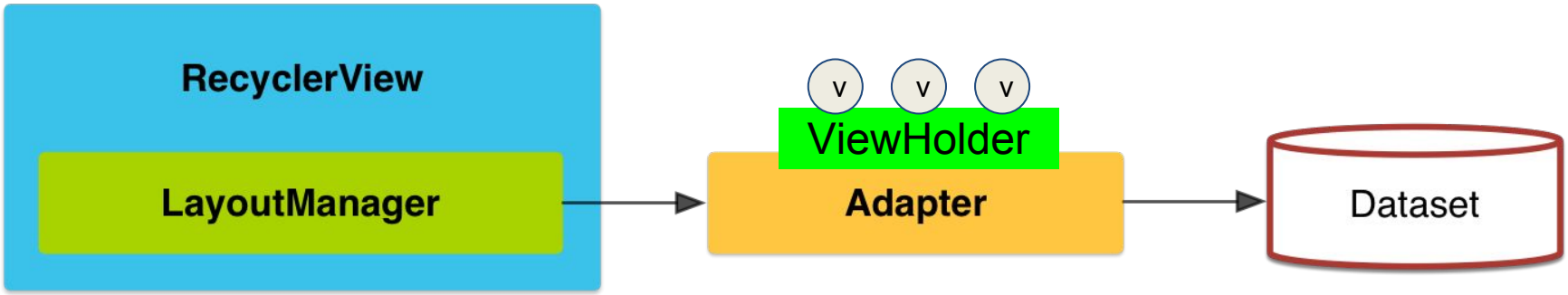
INSTITUTO FEDERAL

RecyclerView

- **RecyclerView** é o componente atualmente indicado para a apresentação de listas
- Possui um nível de responsabilidade menor quando comparado ao ListView
  - Terceiriza a criação do layout de cada item
  - Terceiriza a organização do container de itens

RecyclerView

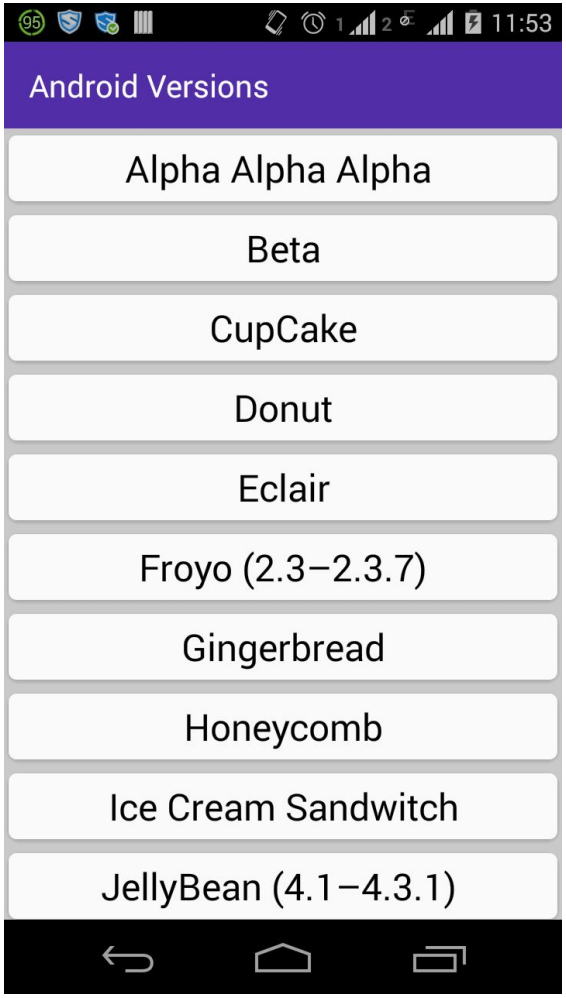
- O RecyclerView encapsula:
  - LayoutManager
  - ItemAnimator
- LayoutManager se comunica com um adapter
  - RecyclerView.Adapter
    - E este acessa/gerencia um conjunto de dados



https://developer.android.com/training/material/lists-cards.html

INSTITUTO FEDERAL

RecyclerView



INSTITUTO FEDERAL

Usando o RecyclerView

- Adicionar dependência no build.gradle

```
dependencies {
    // ...
    implementation 'androidx.recyclerview:recyclerview:1.0.0'
}
```

- Nas versões mais recentes do Android Studio
  - Arraste o RecyclerView no editor de layout
    - Será sugerida a instalação da biblioteca
- Arraste um para o XML da Activity

Adicionando um CardView

- Adicionar dependência no build.gradle

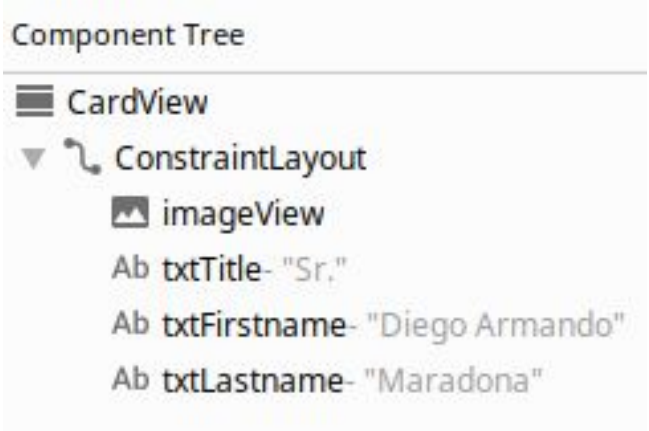
```
dependencies {
    // ...
    implementation androidx.cardview:cardview:1.0.0'
}
```

- Nas versões mais recentes do Android Studio
  - Arraste o CardView no editor de layout
    - Será sugerida a instalação da biblioteca

INSTITUTO FEDERAL

Layout dos Itens

layout/item_user.xml



INSTITUTO FEDERAL

Nosso model: User

- Classe User com suporte ao Room

```
@Entity(tableName = "users")
data class User(
    @ColumnInfo(name = "first_name")
    var firstName: String,
    @ColumnInfo(name = "last_name")
    var lastName: String,
    var title: String
) {
    @PrimaryKey(autoGenerate = true)
    var id: Int = 0
}
```

INSTITUTO FEDERAL

Adapter

- Um **`Adapter`** é quem fornecerá os dados para o RecyclerView
  - Ele utiliza um **`ViewHolder`**
    - Detém a view de cada item a ser mostrado
    - Melhora a performance das listas
    - Cria um cache das views
    - Evita algumas operações de `findViewById()`
      - Impactam muito na fluidez do scroll

INSTITUTO FEDERAL

RecyclerView.Adapter

- Precisamos estender `RecyclerView.Adapter`
- Somos obrigados a implementar 3 métodos
  - `getItemCount()`
    - Retorna total de elementos na listagem
  - `onCreateViewHolder()`
    - Retorna nova instância de viewHolder
  - `onBindViewHolder()`
    - Vincula os dados a uma view

RecyclerView.Adapter

- A assinatura do RecyclerView.Adapter é:

```
public abstract static class Adapter<VH extends ViewHolder>
```

- Portanto, também precisamos criar uma classe que estenda de `ViewHolder`
  - Faremos isto com uma Nested Class

```
class UserAdapter(private val users: List<User>) :
    RecyclerView.Adapter<UserAdapter.ViewHolder>() {

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int) =
        ViewHolder(
            LayoutInflater
                .from(parent.context)
                .inflate(R.layout.user_item, parent, false)
        )
    override fun getItemCount() = users.size

    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        val user = users[position]
        holder.preencherView(user)
    }

    class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        fun preencherView(user: User) {
            itemView.txtTitle.text = user.title
            itemView.txtFirstname.text = user.firstName
            itemView.txtLastname.text = user.lastName
        }
    }
}
```

INSTITUTO FEDERAL

LayoutManager

- O RecyclerView não tem a responsabilidade de posicionar elementos em sua lista
  - Ele delega para um **`LayoutManager`**
  - Isto permite o uso de diferentes layouts
- Três layouts nativos:
  - `LinearLayoutManager`
    - Para listas verticais ou horizontais
  - `GridLayoutManager`
    - Grids
  - `StaggeredGridLayoutManager`
    - Grids com itens de tamanhos variados (mosaico)

INSTITUTO FEDERAL

MainActivity

- Na classe da Activity
  - Definir o LayoutManager do RecyclerView
  - Criar lista de objetos
    - Ou carregar de BD, Web Service, …
  - Instanciar um Adapter
    - Usando lista de objetos criados
  - Atribuir o Adapter ao RecyclerView

```
class MainActivity : AppCompatActivity() {
    lateinit var db: AppDatabase
    lateinit var dao: UserDao
    lateinit var adapter: UserAdapter

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

        db = // ... Room
        dao = db.userDao()

        configureRecyclerView()
        loadUsers()
    }

    private fun loadUsers() {
        adapter = UserAdapter(dao.getAll())
        listUsers.adapter = adapter
    }

    fun configureRecyclerView() {
        listUsers.layoutManager = LinearLayoutManager(
            this, RecyclerView.VERTICAL, false)
    }
}
```

INSTITUTO FEDERAL

Mudando o Layout

- Usando um GridLayoutManager

```
val layout = GridLayoutManager(this, 2)
layout.orientation = RecyclerView.VERTICAL
listUsers.layoutManager = layout
```

- Usando um StaggeredGridLayoutManager

```
val layout = StaggeredGridLayoutManager(
   2, StaggeredGridLayoutManager.VERTICAL)
listUsers.layoutManager = layout
```

INSTITUTO FEDERAL

Referências

- https://medium.com/collabcode/criando-lista-com-recyclerview-no-android-com-kotlin-85cb76f3775d
- https://medium.com/android-dev-br/adeus-listview-recyclerview-ce87e556444d
- http://android-pratap.blogspot.com.br/2015/01/using-linearlayoutmanager.html
- https://developer.android.com/training/material/lists-cards.html
- LECHETA, Ricardo R. Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK. 5ª Edição. Novatec, 2015.