RecyclerView

RecyclerView Configuración

Gradle:

```
dependencies
{

    implementation 'androidx.recyclerview:recyclerview:1.1.0'

}
```

RecyclerView

Definición

res/layout:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
             android:layout_width="match_parent"
             android:layout_height="match_parent">

  <androidx.recyclerview.widget.RecyclerView android:id="@+id/rv_countries"
                                             android:scrollbars="vertical"
                                             android:layout_width="match_parent"
                                             android:layout_height="match_parent"/>

</FrameLayout>
```

Code:

```
val rvCountries : RecyclerView = findViewById<RecyclerView>(R.id.rv_countries);
```

RecyclerView

```
class MainActivity : AppCompatActivity()
{
  lateinit var countries: List<Country>
  lateinit var rvCountries : RecyclerView
  lateinit var countriesAdapter : CountriesAdapter


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

    countries = getCountries()
    rvCountries = findViewById(R.id.rv_countries)

    val layoutManager = LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false)
    rvCountries.layoutManager = layoutManager

    countriesAdapter = CountriesRecyclerViewAdapter(countries)
    rvCountries.adapter = countriesAdapter
  }
```

RecyclerView

RecyclerView.LayoutManager

A través de un **RecyclerView.LayoutManager** se establece cómo se mostrarán los elementos en el RecyclerView

`LinearLayoutManager`
`GridLayoutManager`
`StaggeredGridLayoutManager`

Code:

```
val rvCountries : RecyclerView = findViewById<RecyclerView>(R.id.rv_countries);
val layoutManager : RecyclerView.LayoutManager = …
rvCountries.layoutManager = layoutManager;
```

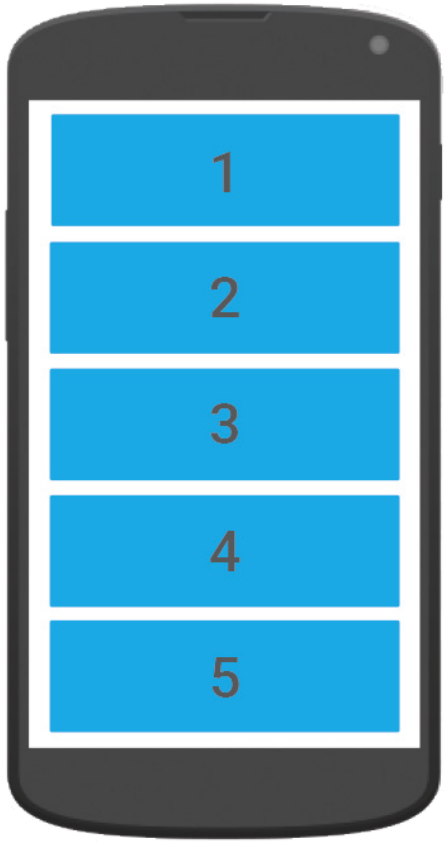
RecyclerView

RecyclerView.LayoutManager

0 1 2 3 4

LinearLayoutManager(Context context, int orientation, boolean reverseLayout)

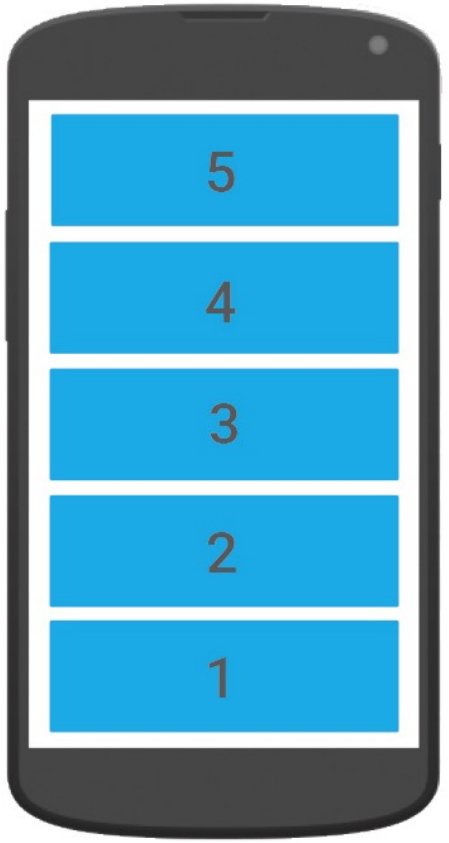
```
val layoutManager : RecyclerView.LayoutManager = LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false)
rvCountries.layoutManager = layoutManager
```



reverseLayout = false

orientation

LinearLayoutManager.*VERTICAL*



reverseLayout = true

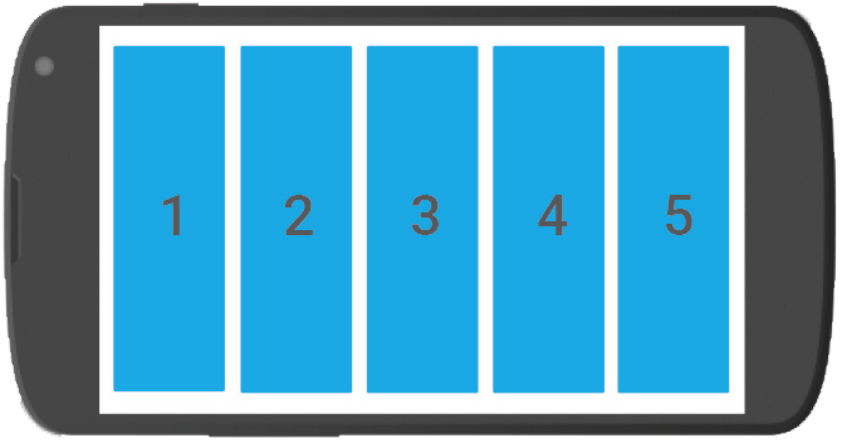
RecyclerView

RecyclerView.LayoutManager

0 1 2 3 4

LinearLayoutManager(Context context, int orientation, boolean reverseLayout)

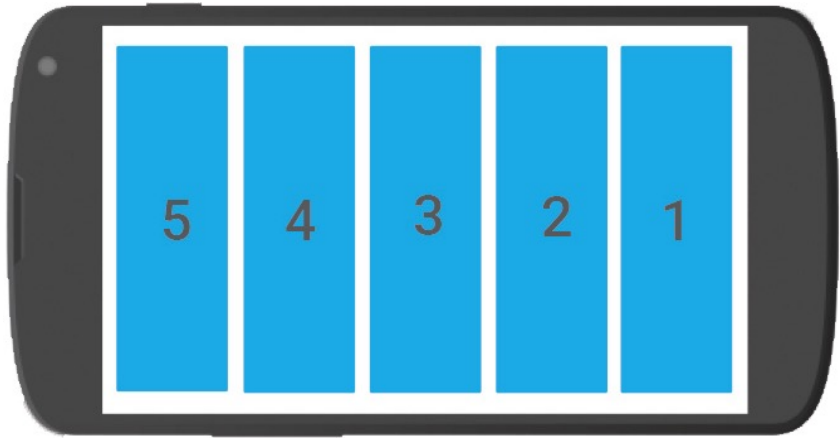
```
val layoutManager : RecyclerView.LayoutManager = LinearLayoutManager(this, LinearLayoutManager.HORIZONTAL, false)
rvCountries.layoutManager = layoutManager
```



reverseLayout = false

orientation

LinearLayoutManager.*HORIZONTAL*



reverseLayout = true

RecyclerView

RecyclerView.LayoutManager

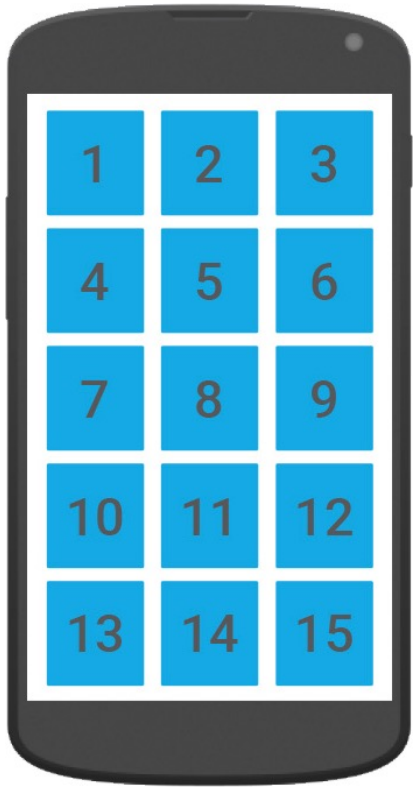

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rv"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
    android:orientation="vertical"
                        vertical
                        horizontal
                        Press ⏎ to insert, ⇥ to replace

    app:reverseLayout="false"
                        false
                        @android:
                        true
                        Press ⏎ to insert, ⇥ to replace
/>
```

RecyclerView

RecyclerView.LayoutManager

GridLayoutManager(Context context, int spanCount, int orientation, boolean reverseLayout)

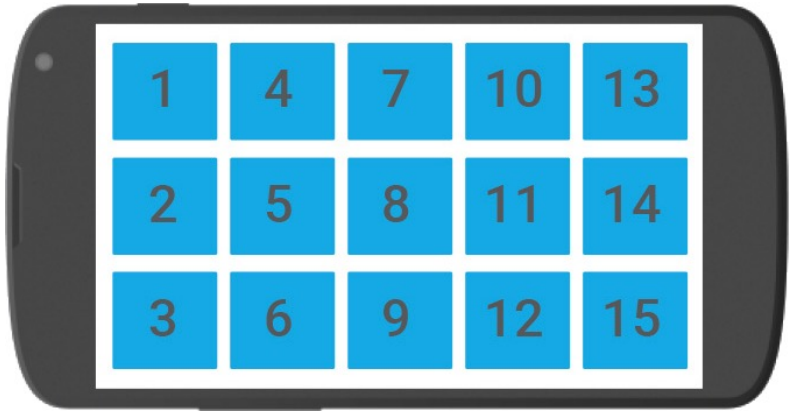
```
val layoutManager RecyclerView.LayoutManager = GridLayoutManager(this, 3, LinearLayoutManager.VERTICAL, false)
rvCountries.layoutManager = layoutManager
```



spanCount=3

orientation
LinearLayoutManager.***VERTICAL***

spanCount=5



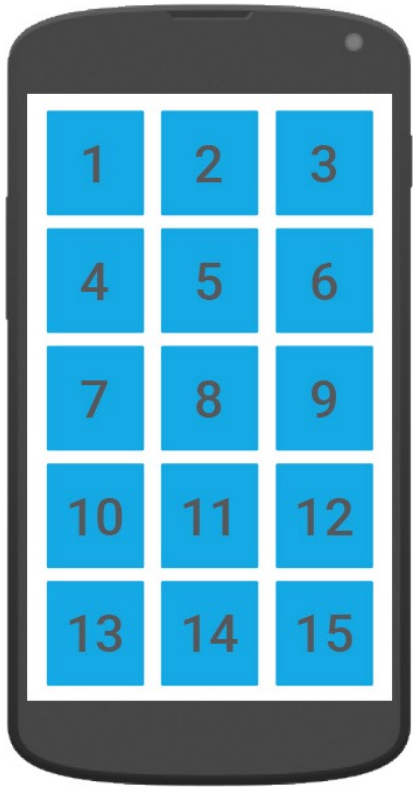
orientation
LinearLayoutManager.***HORIZONTAL***

RecyclerView

RecyclerView.LayoutManager


```
GridLayoutManager(Context context, int spanCount, int orientation, boolean reverseLayout);
```

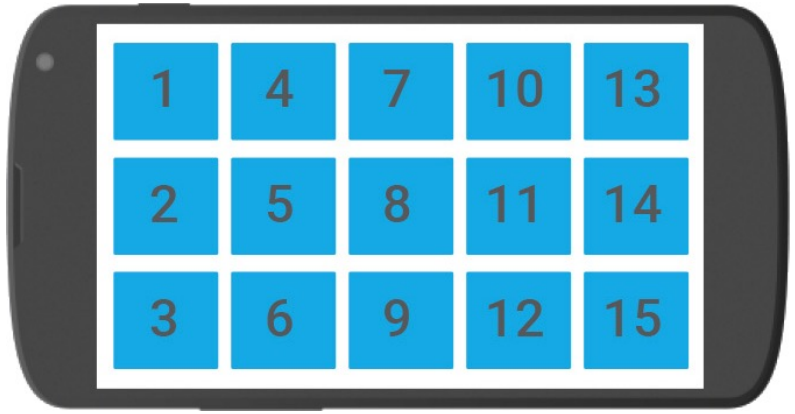
```
val layoutManager RecyclerView.LayoutManager = GridLayoutManager(this, 5, LinearLayoutManager.VERTICAL, false)
rvCountries.layoutManager = layoutManager
```



spanCount=3

orientation
LinearLayoutManager.***VERTICAL***

spanCount=5



orientation
LinearLayoutManager.***HORIZONTAL***

RecyclerView

RecyclerView.LayoutManager


```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rv"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layoutManager="androidx.recyclerview.widget.GridLayoutManager"
    android:orientation="vertical"
    app:spanCount="3"


    app:reverseLayout="false"

 />
```

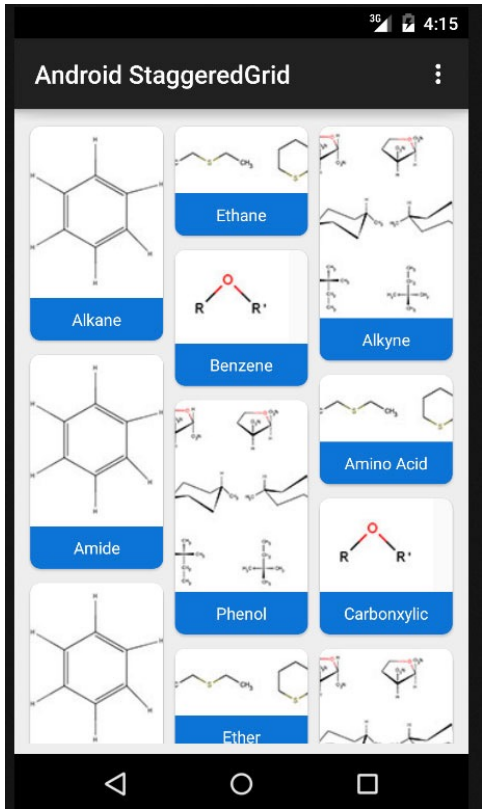
RecyclerView

RecyclerView.LayoutManager

0 1 2 3 **4**

StaggeredGridLayoutManager(int spanCount, int orientation);

```
val layoutManager : RecyclerView.LayoutManager = StaggeredGridLayoutManager(3, LinearLayoutManager.VERTICAL);
rvCountries.layoutManager = layoutManager;
```



RecyclerView

RecyclerView.LayoutManager


```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rv"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layoutManager="androidx.recyclerview.widget.StaggeredGridLayoutManager"
    android:orientation="vertical"
    app:spanCount="3"


    app:reverseLayout="false"

 />
```

RecyclerView

```
class MainActivity : AppCompatActivity()
{
  lateinit var countries: List<Country>
  lateinit var rvCountries : RecyclerView
  lateinit var countriesAdapter : CountriesAdapter


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

    countries = getCountries()
    rvCountries = findViewById(R.id.rv_countries)

    val layoutManager = LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false)
    rvCountries.layoutManager = layoutManager

    countriesAdapter = CountriesRecyclerViewAdapter(countries)
    rvCountries.adapter = countriesAdapter
  }
```

RecyclerView RecyclerView.Adapter

A través de un **RecyclerView.Adapter** se establece los datos subyacentes que debe mostrar el RecyclerView

Code:

```
val rvCountries : RecyclerView = findViewById<RecyclerView>(R.id.rv_countries)
Val adapter : RecyclerView.Adapter = …
rvCountries.adapter = adapter
```

RecyclerView

RecyclerView.Adapter

```
class CountriesAdapter : RecyclerView.Adapter< CountriesAdapter.CountryViewHolder >()
{


  class CountryViewHolder(view: View) : RecyclerView.ViewHolder(view)
  {

  }

}
```

Debe ser un RecyclerView.ViewHolder

RecyclerView

RecyclerView.Adapter

```
class CountriesAdapter : RecyclerView.Adapter< CountriesAdapter.CountryViewHolder >()
{

    override fun onCreateViewHolder(parent : ViewGroup, viewType : Int) : CountryViewHolder
    {
    }


    override fun onBindViewHolder(holder: CountryViewHolder, position: Int)
    {
    }


    override fun getItemCount(): Int
    {
    }

    class CountryViewHolder(view: View) : RecyclerView.ViewHolder(view)
    {

    }
}
```

Crea un nuevo RecyclerView.ViewHolder

RecyclerView

RecyclerView.Adapter

```
class CountriesAdapter : RecyclerView.Adapter< CountriesAdapter.CountryViewHolder >()
{

    override fun onCreateViewHolder(parent : ViewGroup, viewType : Int) : CountryViewHolder
    {
    }


    override fun onBindViewHolder(holder: CountryViewHolder, position: Int)
    {
    }


    override fun getItemCount(): Int
    {
    }

    class CountryViewHolder(view: View) : RecyclerView.ViewHolder(view)
    {

    }
}
```

Reutiliza un RecyclerView.ViewHolder con el que mostrar el elemento position

RecyclerView

RecyclerView.Adapter

```
class CountriesAdapter : RecyclerView.Adapter< CountriesAdapter.CountryViewHolder >()
{

    override fun onCreateViewHolder(parent : ViewGroup, viewType : Int) : CountryViewHolder
    {
    }


    override fun onBindViewHolder(holder: CountryViewHolder, position: Int)
    {
    }


    override fun getItemCount(): Int
    {
    }

    class CountryViewHolder(view: View) : RecyclerView.ViewHolder(view)
    {

    }
}
```

Indica cuántos elementos mostrará el RecyclerView

RecyclerView

RecyclerView.Adapter

```
class CountriesAdapter(val countries : List<Country>): RecyclerView.Adapter<CountryViewHolder>()
{

  @Override
  public int getItemCount()
  {
    return countries.size();
  }

}
```

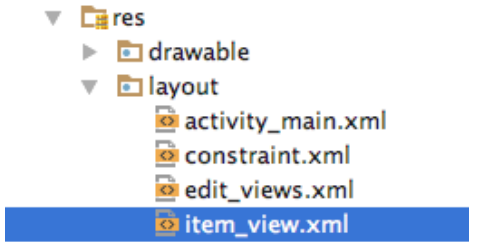
Cuando se crea el RecyclerView.Adapter se pasarán como parámetro los datos subyacentes

RecyclerView

RecyclerView.Adapter

Definición del layout de cada item del RecyclerView

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
                                   xmlns:card_view="http://schemas.android.com/apk/res-auto"
                                   card_view:cardCornerRadius="6dp"
                                   card_view:cardElevation="4dp"
                                   card_view:cardUseCompatPadding="true"
                                   android:layout_width="match_parent"
                                   android:layout_height="wrap_content">

    <LinearLayout android:orientation="vertical"
                  android:layout_width="match_parent”
                  android:layout_height="wrap_content">

      <TextView android:id="@+id/tv_line1"
                android:textAppearance="?android:textAppearanceListItem"
                android:layout_marginBottom="5dp"
                android:layout_width="match_parent”
                android:layout_height="wrap_content"/>

      <TextView android:id="@+id/tv_line2"
                android:textAppearance="?android:textAppearanceListItemSecondary"
                android:layout_width="match_parent”
                android:layout_height="wrap_content"/>
    </LinearLayout>

</android.support.v7.widget.CardView>
```



RecyclerView

RecyclerView.Adapter

```
class CountriesAdapter(val countries: List<Country>) : RecyclerView.Adapter<CountryViewHolder>()
{

  override fun onCreateViewHolder(parent: ViewGroup, viewType: Int) : CountryViewHolder
  {
    val context : Context = parent.context;
    val view : View = LayoutInflater.from(context).inflate(R.layout.item_view, parent, false);
    val countryViewHolder : CountryViewHolder = CountryViewHolder(view);
    return countryViewHolder;
  }

  class CountryViewHolder(view: View) : RecyclerView.ViewHolder(view)
  {
    val tvLine1 : TextView = view.findViewById<TextView>(R.id.tv_line1);
    val tvLine2 : TextView = view.findViewById<TextView>(R.id.tv_line2);



  }
```

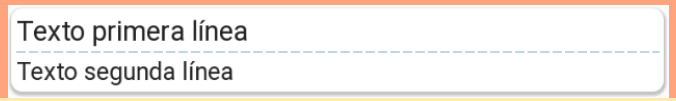
Después de "inflar" la vista con la que mostrar el item, se creará el ViewHolder a partir de ella

RecyclerView

RecyclerView.Adapter

```
class CountriesAdapter(val countries: List<Country>) : RecyclerView.Adapter<CountryViewHolder>()
{

  override fun onCreateViewHolder(parent: ViewGroup, viewType: Int) : CountryViewHolder
  {
    val context : Context = parent.context;
    val view : View = LayoutInflater.from(context).inflate(R.layout.item_view, parent, false);
    val countryViewHolder : CountryViewHolder = CountryViewHolder(view);
    return countryViewHolder;
  }

  class CountryViewHolder(view: View) : RecyclerView.ViewHolder(view)
  {
    val tvLine1 : TextView = view.findViewById<TextView>(R.id.tv_line1);
    val tvLine2 : TextView = view.findViewById<TextView>(R.id.tv_line2);



  }
```

Todo ViewHolder contiene el atributo itemView que referencia la View a la que está asociado



RecyclerView

RecyclerView.Adapter

```
class CountriesAdapter(val countries: List<Country>) : RecyclerView.Adapter<CountryViewHolder>()
{

  override fun onCreateViewHolder(parent: ViewGroup, viewType: Int) : CountryViewHolder
  {
    val context : Context = parent.context;
    val view : View = LayoutInflater.from(context).inflate(R.layout.item_view, parent, false);
    val countryViewHolder : CountryViewHolder = CountryViewHolder(view);
    return countryViewHolder;
  }

  class CountryViewHolder(view: View) : RecyclerView.ViewHolder(view)
  {
    val tvLine1 : TextView = view.findViewById<TextView>(R.id.tv_line1);
    val tvLine2 : TextView = view.findViewById<TextView>(R.id.tv_line2);



  }
```

Cuando se crea el ViewHolder se localizan los widgets que se utilizarán para mostrar los datos del item del RecyclerView

RecyclerView

RecyclerView.Adapter

```
class CountriesAdapter(countries: List<Country>) : RecyclerView.Adapter<CountryViewHolder>()
{
  @Override
  override fun onBindViewHolder(holder: CountryViewHolder, position: Int)
  {
    val country: Country = countries.get(position)
    holder.bind( country )
  }

  class CountryViewHolder(view: View) : RecyclerView.ViewHolder(view)
  {
    val tvLine1 : TextView = …
    val tvLine2 : TextView = …

    fun bind(country : Country)
    {
      tvLine1.text = country.name;
      tvLine2.text = "${country.population)"
    }
  }
}
```

Cuando se tiene que mostrar un item se reutiliza un ViewHolder

RecyclerView

RecyclerView.Adapter

```
class CountriesAdapter(countries: List<Country>) : RecyclerView.Adapter<CountryViewHolder>()
{
  @Override
  override fun onBindViewHolder(holder: CountryViewHolder, position: Int)
  {
    val country: Country = countries.get(position)
    holder.bind( country )
  }

  class CountryViewHolder(view: View) : RecyclerView.ViewHolder(view)
  {
    val tvLine1 : TextView = …
    val tvLine2 : TextView = …

    fun bind(country : Country)
    {
      tvLine1.text = country.name;
      tvLine2.text = "${country.population)"
    }
  }
}
```

Después de localizar el dato subyacente, se muestran los datos en los widgets correspondientes