

## RECYCLERVIEWS: Segundo ejemplo RecyclerView, más completo. Planetas.

En esta app, utilizaremos otra RecyclerView, más completa que en el ejemplo inicial. Aquí ya no hay explicaciones, sino los pasos para ir siguiendo la implementación de esta app.

Crea un proyecto: File→New Project → Empty Views Activity, lo llamaremos, por ejemplo, **RecyclerViewAvanzado**.

Consideraciones previas:

En el paquete modelo, crearemos las siguientes clases:

### Clase Planet

```
package modelo

data class Planet(val name: String, val sizeKm: Int, val distanceAU: Double, val
imageName: String = name.lowercase())
```

### Clase PlanetData

```
package modelo

object PlanetData {
    private val planets = mutableListOf(
        Planet("Mercurio", 4879, 0.39, "mercurio"),
        Planet("Venus", 12104, 0.72, "venus"),
        Planet("Tierra", 12756, 1.00, "tierra"),
        Planet("Marte", 6792, 1.52, "marte"),
        Planet("Jupiter", 142984, 5.20, "jupiter"),
        Planet("Saturno", 120536, 9.58, "saturno"),
        Planet("Urano", 51118, 19.22, "urano"),
        Planet("Neptuno", 49528, 30.05, "neptuno")
    )

    fun getPlanets(): MutableList<Planet> {
        return planets
    }
}
```

1. En la actividad principal se ha añadido el RecyclerView.
2. Crea el archivo xml, para dar aspecto a cada ítem. Se debe mostrar el nombre, tamaño, distancia y la imagen del planeta (que se llama igual que el nombre del planeta), lo llamaremos **item\_planeta.xml**. Para crear un xml con constraint layout se hace lo siguiente: New→XML→Layout XML File  
En **Root Element**, usa `androidx.constraintlayout.widget.ConstraintLayout` u otro layout
3. Dentro del paquete adaptador, crea **PlanetViewHolder** con las variables asociadas a las views del ítem
4. Crea el adaptador del RecyclerView que llamaremos **PlanetAdapter** dentro del paquete adaptador.
  - a. Se va almacenando el índice del elemento que se pulsa, para así poder mostrarlo en un Toast, o como veremos, gracias a ese índice pasaremos el detalle a otra ventana.
  - b. Si se hace pulsación larga, sale un Alert Dialog indicando si realmente queremos eliminar
  - c. Se ha añadido un método, `getSelectedItems` (para así saber qué elementos han quedado seleccionados al pulsar el botón detalle en el activity principal)
  - d. Se controla si se pulsa otra vez en un elemento y se le deja el color inicial
5. Se ha añadido un botón detalle en el activity principal. AL pulsar en él, vamos a un segundo activity llamado **DetallePlanetas**, y mostramos el detalle de los planetas seleccionados.