

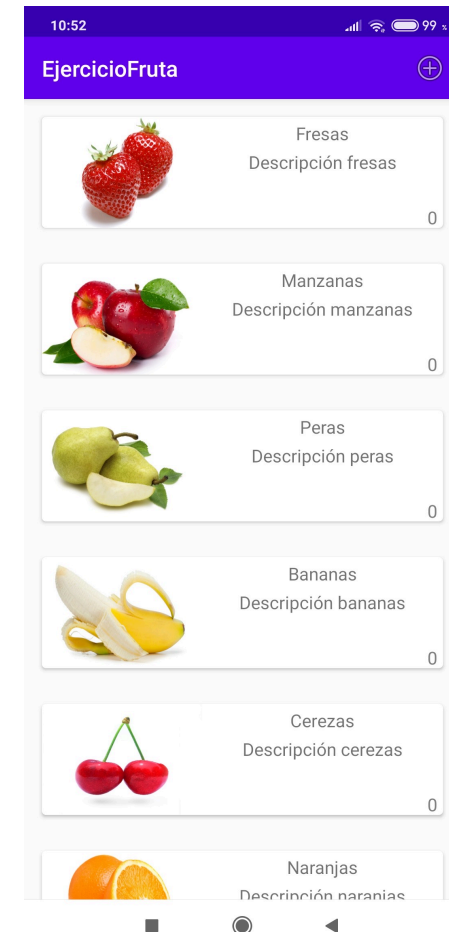
Programación Android

Ejercicio

Crear una app con un **RecyclerView** usando **CardView** para sus ítems. Cada CardView, debe tener una imagen a su izquierda, ocupando el **40%** del CardView, y dejando un **60%** para el resto. La altura será de 100dp.

Nuestro modelo **Fruta**, deberá tener las siguientes propiedades: nombre, descripción, imagen para el background y cantidad. Establecer el límite de cantidad a 10, y el valor para resetear a 0.

En la parte del 40%, mostramos la imagen de background de nuestro modelo, y en la del 60%, enseñaríamos, nombre, descripción y cantidad.



Continúa en la siguiente página

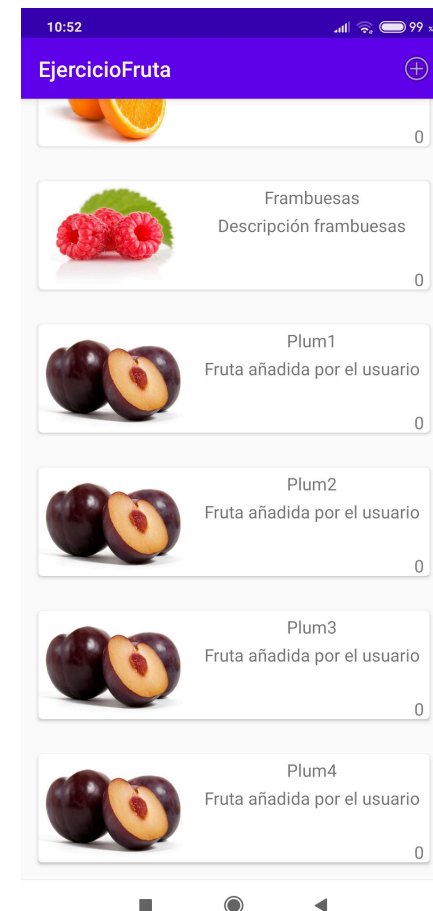
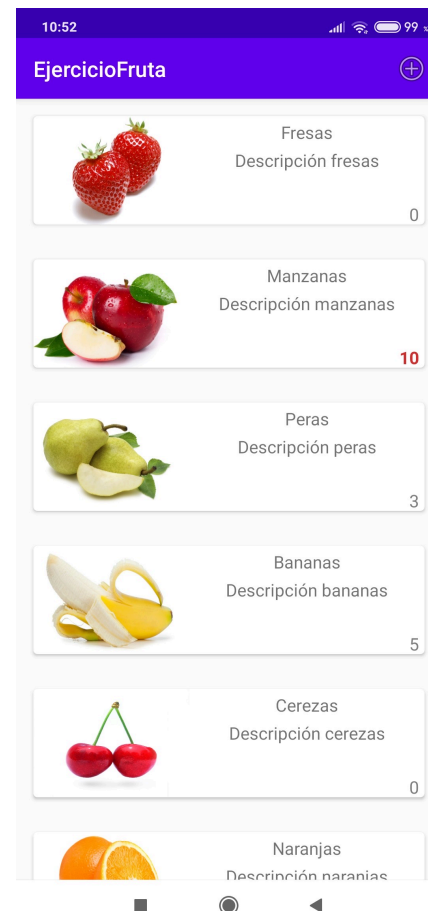


Programación Android

Ejercicio

Crear un **botón en el menú de opciones** para añadir una nueva fruta a nuestro RecyclerView. Las nuevas frutas pueden tener un contador en el nombre (Plum1, Plum2...) para diferenciarlas entre sí, aunque los demás valores del modelo pueden repetirse entre las nuevas frutas.

Para añadir una pieza a la cantidad total de cada fruta, **pulsaremos en la imagen de background** que ocupa la parte izquierda de nuestro ítem.



Continúa en la siguiente página



Programación Android

Ejercicio

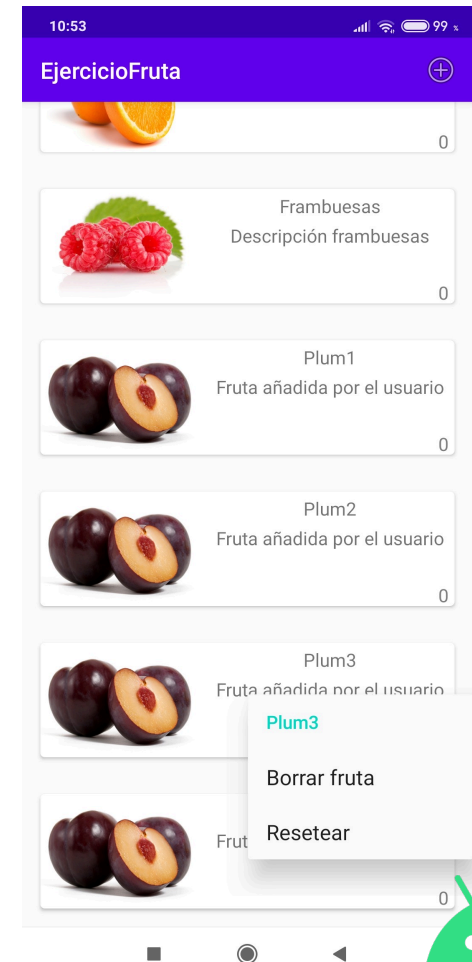
Crear un **Context Menu** para nuestros elementos en el RecyclerView, con 2 opciones principales:

- Borrar: borrar el elemento en sí
- Resetear cantidad: establece la cantidad a 0

El **Context Menu** debe mostrar como título el nombre de la fruta. Ayúdate de **Picasso** para la carga del background de cada ítem.

***Pista:** es posible que para implementar el Context Menu sea necesario investigar un poco, ya que no se comporta igual que el ListView en estos términos. Posiblemente, una solución válida sería implementarlo dentro de nuestro ViewHolder en nuestro Adapter.*

Recuerda crear 3 paquetes para manejar los diferentes activities, adapters, models.



Programación Android

Ejercicio

Para implementar el menú contextual tendremos que hacerlo en la clase **ViewHolder**

En primer lugar crearemos un bloque init para inicializar el atributo itemView que recibe la clase, justo detrás de los atributos de la clase ViewHolder:

```
init {  
    itemView.setOnCreateContextMenuListener(this)  
}
```

Y después, dentro de esta clase, implementaremos los métodos onCreateContextMenu y onOptionsItemSelected

```
override fun onCreateContextMenu(  
    menu: ContextMenu?,  
    v: View?,  
    menuInfo: ContextMenu.ContextMenuInfo?  
) {  
    val frutaSeleccionada: Fruta = frutas[adapterPosition]  
    menu!!.setHeaderTitle(frutaSeleccionada.nombre)  
    val inflater: MenuInflater = activity.menuInflater  
    inflater.inflate(R.menu.context_menu_fruta, menu)  
    for (i in 0 until menu.size()) {  
        menu.getItem(i).setOnMenuItemClickListener(this)  
    }  
}
```

```
override fun onOptionsItemSelected(item: MenuItem?): Boolean {  
    return when (item?.itemId) {  
        R.id.delete_fruta -> {  
            frutas.removeAt(adapterPosition)  
            notifyItemRemoved(adapterPosition)  
            true  
        }  
        R.id.reset_fruta -> {  
            frutas[adapterPosition].reset()  
            notifyItemChanged(adapterPosition)  
            false  
        }  
        else -> false  
    }  
}
```

