

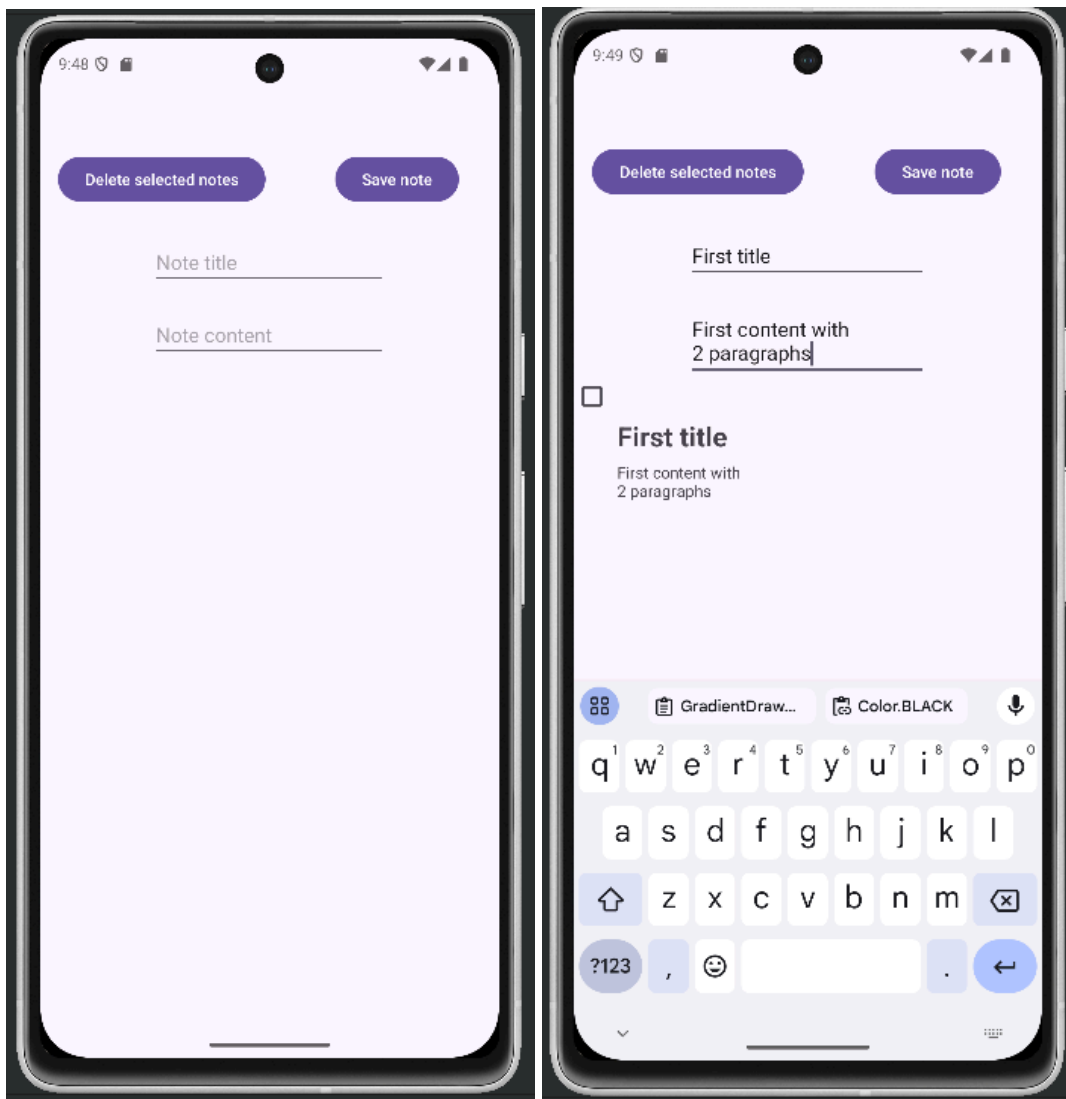
## Tarea individual 6 - Notas Rápidas

Se inicia la actividad con dos botones: Uno para borrar las notas seleccionadas, y otro para guardar una nota.

No genera error ninguna de las dos aunque no haya ni notas guardadas para borrar, ni contenido en los EditText.

Así, al crear una nota, genera un layout, con un CheckBox, y dos TextViews. El primer TextView está en negrita y tiene el tamaño de texto más grande.

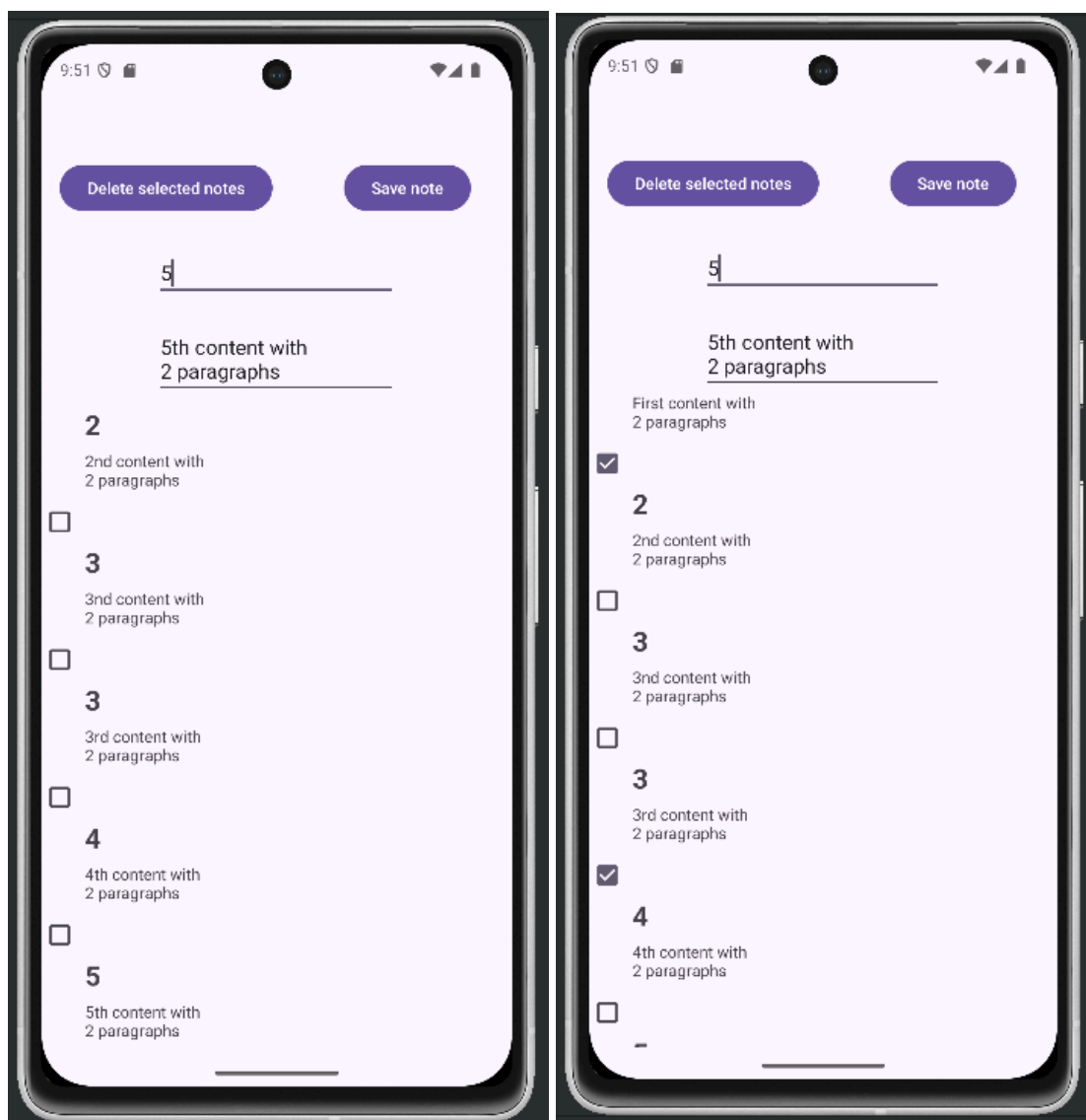
El segundo EditText es multiline, por lo que puede generar strings con saltos de línea.



El contenedor de layouts en el que se almacena cada nota está dentro de un ScrollView, por lo que se pueden generar las notas que queramos y se podrán ver.

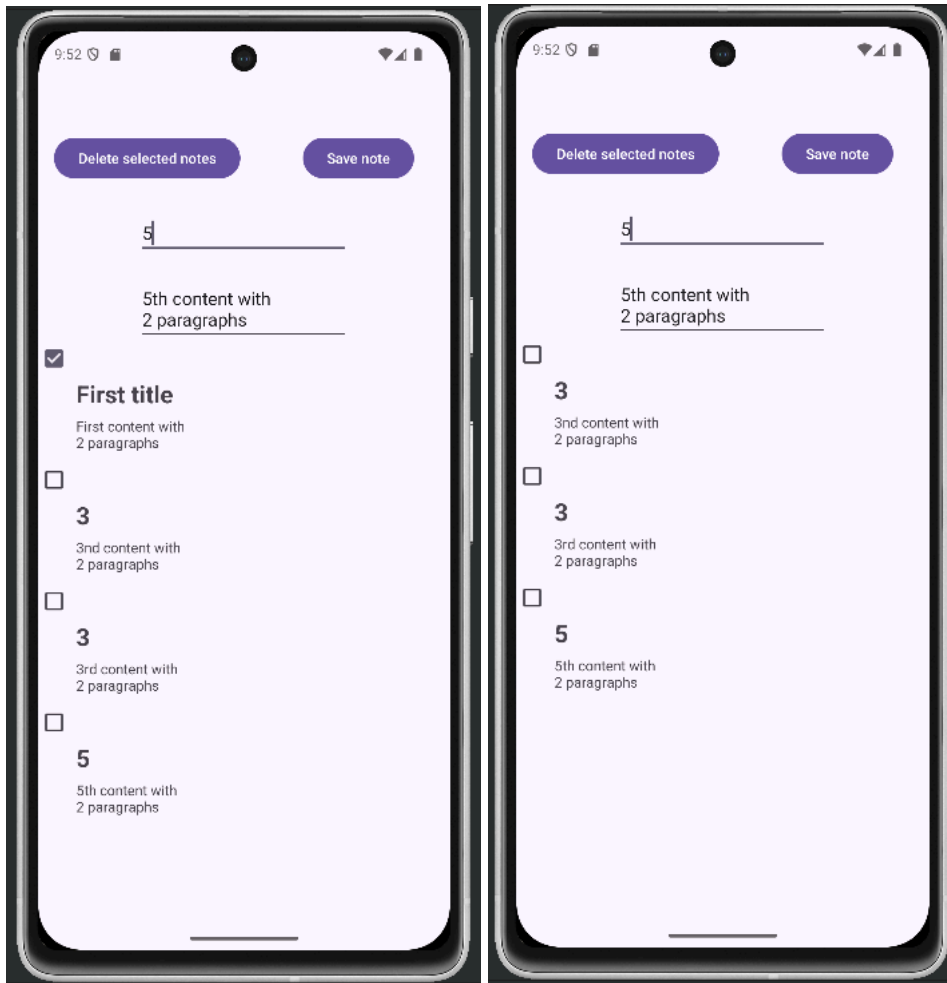
Los checkbox tienen un 'checkedChangeListener' para avisar cuando son seleccionados.

A cada layout se le da una id(par) y a cada checkbox se le da otra(impar) y se almacena en un array booleano los que están seleccionados.



Así, cuando se borra uno seleccionado, lo que hace es mirar en el booleano dónde tiene 'true', y borra el view anterior, que será el layout completo de esa nota. Como a cada nota se genera un id diferente, se podrían generar notas infinitas, subiendo la capacidad del array contenedor de IDs.

Cuando le das a borrar, el bucle recorre todo el array buscando un true, y lo cambia a false para que al volver a dar al botón no vuelva a intentar borrar una View ya borrada.



Código:

```

public class MainActivity extends AppCompatActivity {

    private Button btnSaveNote, btnDelete; 2 usages
    private LinearLayout linearLayout; 3 usages
    private EditText etNoteTitle, etmlNoteContent; 2 usages

    private boolean[] itemsSelected; 6 usages
    private int numIds = 0; //even for layouts and odd for checkbox 4 usages

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable( $this$enableEdgeToEdge: this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars);
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        itemsSelected = new boolean[100];

        etNoteTitle = findViewById(R.id.etNoteTitle);
        etmlNoteContent = findViewById(R.id.etmlNoteContent);
        linearLayout = findViewById(R.id.linearLayout);

```

```

47         btnSaveNote = findViewById(R.id.btnSaveNote);
48         btnSaveNote.setOnClickListener(new View.OnClickListener(){
49             @Override
50             public void onClick(View v){
51
52                 String noteTitle = String.valueOf(etNoteTitle.getText());
53                 String noteContent = String.valueOf(etmlNoteContent.getText());
54
55                 LinearLayout ll = new LinearLayout( context: MainActivity.this);
56                 ll.setId(numIds); numIds ++;
57                 TextView tv = new TextView( context: MainActivity.this);
58                 tv.setPadding( left: 100, top: 8, right: 8, bottom: 8);
59                 tv.setTypeface( tf: null, Typeface.BOLD);
60                 tv.setTextSize(24);
61                 TextView TextViewML = new TextView( context: MainActivity.this);
62                 TextViewML.setPadding( left: 100, top: 8, right: 8, bottom: 30);
63
64                 tv.setText(noteTitle);
65                 TextViewML.setText(noteContent);
66
67                 CheckBox ck = new CheckBox( context: MainActivity.this);
68                 ck.setId(numIds); numIds++;
69                 int checkBoxId = ck.getId();

```

```

70         ck.setPadding( left: 150, top: 0, right: 0, bottom: 0);
71         ck.setOnCheckedChangeListener((buttonView, isChecked) -> {
72             int actualId = ck.getId();
73
74             if (isChecked) {
75                 itemsSelected[actualId] = true;
76
77             } else {
78                 itemsSelected[actualId] = false;
79             }
80         });
81
82         ll.addView(ck);
83         ll.addView(tv);
84         ll.addView(TextviewML);
85         LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(
86             LinearLayout.LayoutParams.WRAP_CONTENT,
87             LinearLayout.LayoutParams.WRAP_CONTENT
88         );
89         ll.setLayoutParams(params);
90         ll.setOrientation(LinearLayout.VERTICAL);
91
92         linearLayout.addView(ll);
93     });
94
95     btnDelete = findViewById(R.id.btnDelete);
96     btnDelete.setOnClickListener(new View.OnClickListener(){
97         @Override
98         public void onClick(View v){
99
100             for(int x = itemsSelected.length - 1 ; x>=0; x--){
101
102                 if(itemsSelected[x] == true){
103                     linearLayout.removeView(findViewById(x-1));
104                     itemsSelected[x] = false;
105                 }
106             }
107         }
108     });
109 }
110 }

```