“FUNDAMENTOS DE KOTLIN”

Resultados de Aprendizaje:

- Comprender los elementos necesarios para iniciarse en el desarrollo de aplicaciones basadas en Android.
- Desarrollar pequeños proyectos haciendo uso de los elementos que proporciona Android.

INDICE

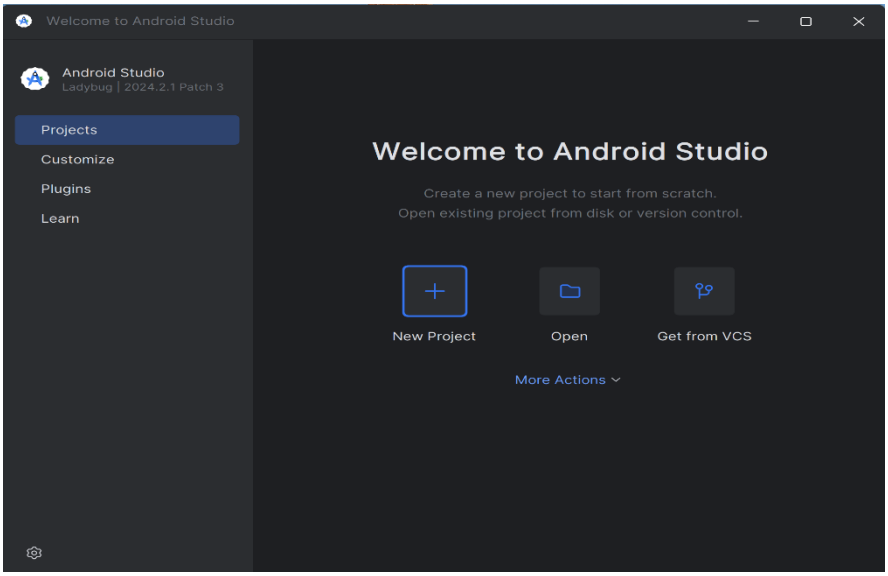
INTRODUCCIÓN

Para desarrollar esta práctica usted deberá tener los siguientes paquetes de software:

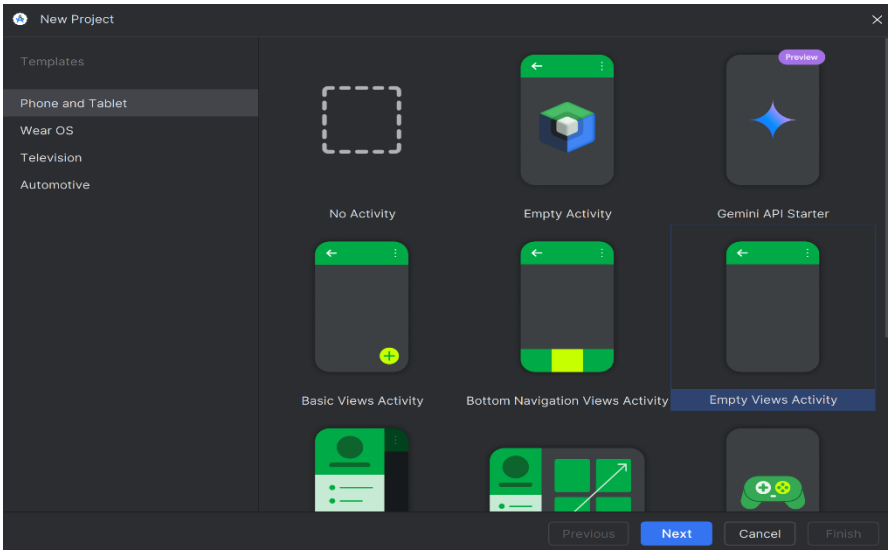
- Android Studio Versión 4 o superior
- Equipo Móvil con Android (Puede ser Tablet o Smartphone) con **modo desarrollador habilitado.**
  (Ver https://www.youtube.com/watch?v=wLJS8CKo95o)

CREANDO NUESTRO PROYECTO

1. Iniciamos nuestro Android Studio.



2. Elegimos la plantilla **Empty Views Activity.**

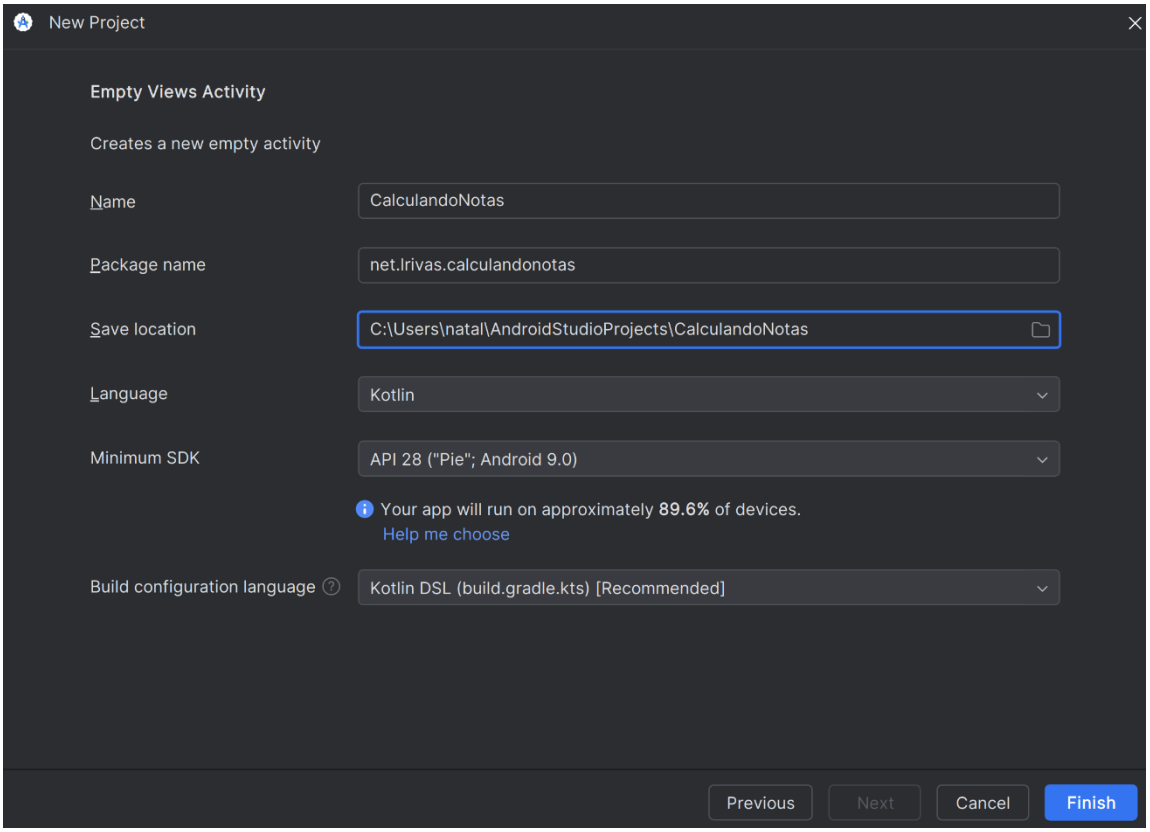


3. Luego establezca las siguientes propiedades al proyecto:
   a. Nombre: **CalculandoNotas**

b. Package Name: **net.lrivas.calculandonotas**

c. Lenguaje: **Kotlin**

d. Minimum Api: **28**



4. Una vez que se genere el proyecto, de clic sobre **activity_main.xml**. Y copie y pegue el siguiente contenido:

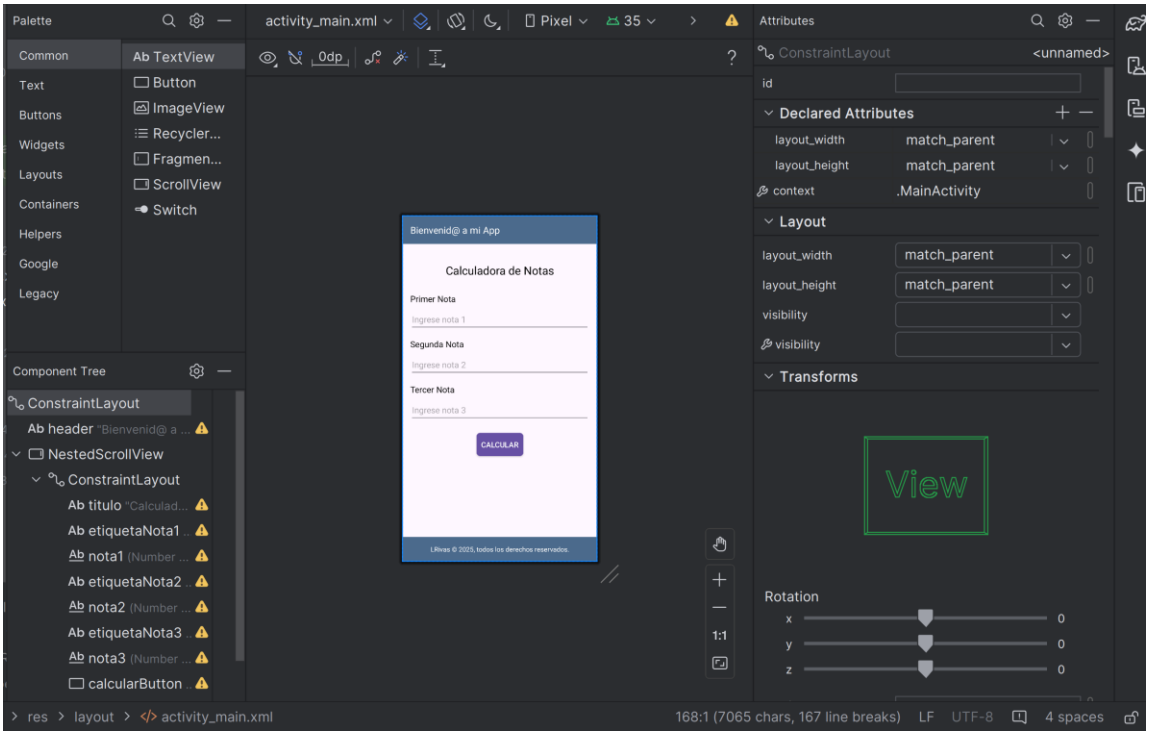
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/header"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Bienvenid@ a mi App"
        android:textSize="20sp"
        android:textColor="@color/white"
        android:background="@color/colorPrimary"
        android:gravity="start"
        android:padding="16dp"
        app:layout_constraintTop_toTopOf="parent"/>

    <androidx.core.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="0dp"
```

```
        android:fillViewport="true"
        app:layout_constraintTop_toBottomOf="@id/header"
        app:layout_constraintBottom_toTopOf="@id/footer">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="16dp">

            <TextView
                android:id="@+id/titulo"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Calculadora de Notas"
                android:textSize="24sp"
                android:textColor="@color/black"
                android:layout_marginTop="24dp"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintEnd_toEndOf="parent" />

            <!-- Etiqueta para la primera nota -->
            <TextView
                android:id="@+id/etiquetaNota1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="32dp"
                android:text="Primer Nota"
                android:textColor="@color/black"
                android:textSize="18sp"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@id/titulo" />

            <EditText
                android:id="@+id/nota1"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:hint="Ingrese nota 1"
                android:inputType="numberDecimal"
                android:layout_marginTop="8dp"

app:layout_constraintTop_toBottomOf="@id/etiquetaNota1"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                android:minHeight="48dp"
                android:importantForAutofill="no" />

            <!-- Etiqueta para la segunda nota -->
            <TextView
                android:id="@+id/etiquetaNota2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="16dp"
                android:text="Segunda Nota"
                android:textColor="@color/black"
                android:textSize="18sp"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@id/nota1" />

            <EditText
```

```
                android:id="@+id/nota2"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:hint="Ingrese nota 2"
                android:inputType="numberDecimal"
                android:layout_marginTop="8dp"

app:layout_constraintTop_toBottomOf="@id/etiquetaNota2"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                android:minHeight="48dp"
                android:importantForAutofill="no" />

            <!-- Etiqueta para la tercera nota -->
            <TextView
                android:id="@+id/etiquetaNota3"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="16dp"
                android:text="Tercer Nota"
                android:textColor="@color/black"
                android:textSize="18sp"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@id/nota2" />

            <EditText
                android:id="@+id/nota3"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:hint="Ingrese nota 3"
                android:inputType="numberDecimal"
                android:layout_marginTop="8dp"

app:layout_constraintTop_toBottomOf="@id/etiquetaNota3"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                android:minHeight="48dp"
                android:importantForAutofill="no" />

            <!-- Botón para calcular -->
            <Button
                android:id="@+id/calcularButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Calcular"
                android:layout_marginTop="24dp"
                style="@style/CustomButtonStyle"
                android:padding="10dp"
                app:layout_constraintTop_toBottomOf="@id/nota3"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                android:minHeight="48dp" />

            <!-- Resultado -->
            <TextView
                android:id="@+id/resultado"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Resultado: "
                android:textSize="18sp"
```

```
                android:textColor="@color/black"

app:layout_constraintTop_toBottomOf="@id/calcularButton"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                android:layout_marginTop="24dp"
                android:visibility="gone" />

        </androidx.constraintlayout.widget.ConstraintLayout>
    </androidx.core.widget.NestedScrollView>

    <!-- Footer (Pie de página) -->
    <TextView
        android:id="@+id/footer"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="LRivas © 2025, todos los derechos reservados."
        android:textSize="14sp"
        android:textColor="@color/white"
        android:background="@color/colorPrimary"
        android:gravity="center"
        android:padding="16dp"
        app:layout_constraintBottom_toBottomOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

5. Deberá mostrarse una interfaz como la que se ve a continuación al momento de cambiar de la vista de **Code** a vista de **Design**.

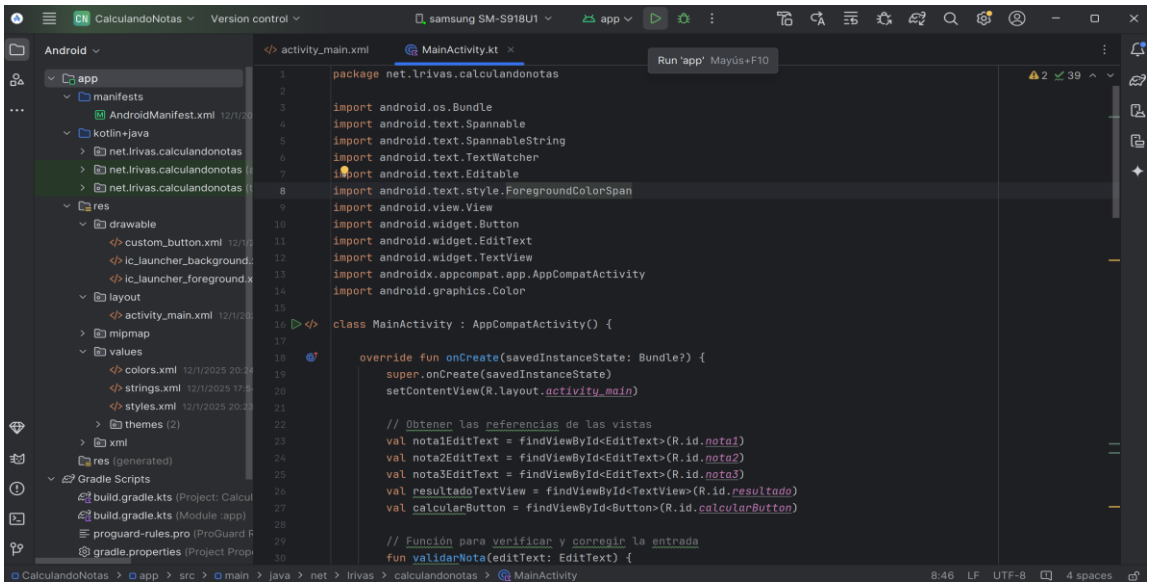


6. Ahora vaya al archivo **MainActivity.kt** y agregue el siguiente código, deberá quedar tal cual como se ve a continuación:

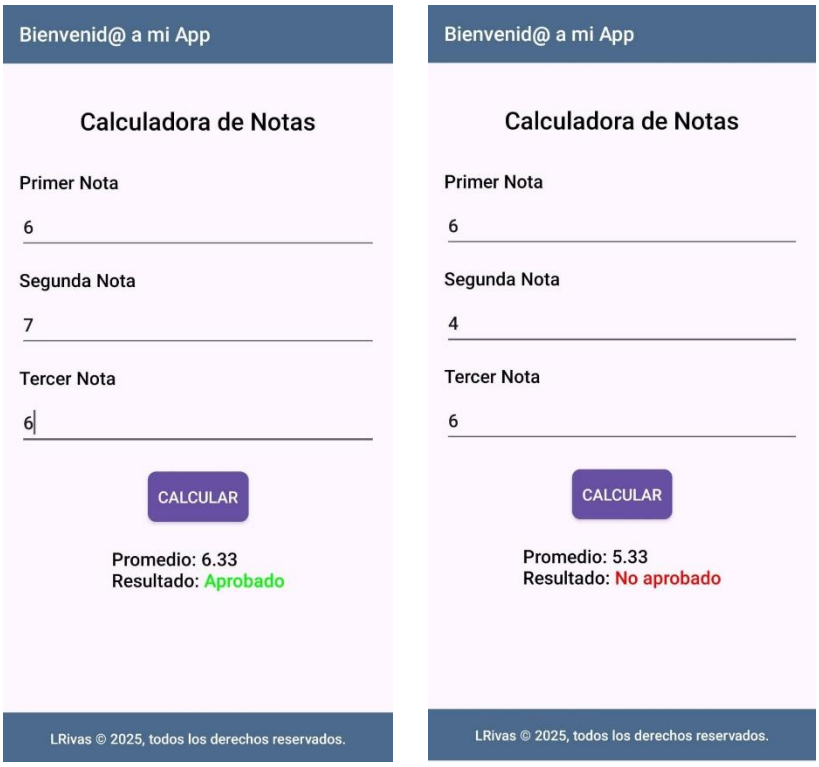
```
package net.lrivas.sensorproximidad


import android.os.Bundle
import android.text.Spannable
import android.text.SpannableString
import android.text.TextWatcher
import android.text.Editable
import android.text.style.ForegroundColorSpan
import android.view.View
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import android.graphics.Color

class MainActivity : AppCompatActivity() {

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

        // Obtener las referencias de las vistas
        val nota1EditText = findViewById<EditText>(R.id.nota1)
        val nota2EditText = findViewById<EditText>(R.id.nota2)
        val nota3EditText = findViewById<EditText>(R.id.nota3)
        val resultadoTextView = findViewById<TextView>(R.id.resultado)
        val calcularButton = findViewById<Button>(R.id.calcularButton)

        // Función para verificar y corregir la entrada
        fun validarNota(editText: EditText) {
            editText.addTextChangedListener(object : TextWatcher {
                override fun afterTextChanged(s: Editable?) {
                    val text = s.toString()

                    // Verificar si la entrada es un número válido
entre 0 y 10
                    if (text.isNotEmpty()) {
                        val input = text.toDoubleOrNull()
                        if (input != null && (input < 0 || input >
10)) {
                            // Si el número está fuera de rango,
ajustar a 10 como máximo
                            editText.setText("10")

editText.setSelection(editText.text.length)  // Mover el cursor al
final
                        } else if (input == null) {
                            editText.setText("")
                        }
                    }
                }

                override fun beforeTextChanged(s: CharSequence?,
start: Int, count: Int, after: Int) {}
```

```
                override fun onTextChanged(s: CharSequence?, start:
Int, before: Int, count: Int) {}
            })
        }

        // Aplicar la validación a todos los campos de nota
        validarNota(nota1EditText)
        validarNota(nota2EditText)
        validarNota(nota3EditText)


        calcularButton.setOnClickListener {
            // Obtener las notas ingresadas por el usuario
            val nota1 = nota1EditText.text.toString().toDoubleOrNull()
            val nota2 = nota2EditText.text.toString().toDoubleOrNull()
            val nota3 = nota3EditText.text.toString().toDoubleOrNull()

            // Verificar si las notas son válidas
            if (nota1 != null && nota2 != null && nota3 != null) {
                val promedio = (nota1 + nota2 + nota3) / 3

                val resultado = if (promedio >= 6) {
                    "Aprobado"
                } else {
                    "No aprobado"
                }

                val resultadoText = "Promedio: %.2f\nResultado:
$resultado".format(promedio, resultado)

                val spannable = SpannableString(resultadoText)

                val colorSpan = if (resultado == "Aprobado") {
                    ForegroundColorSpan(Color.GREEN)
                } else {
                    ForegroundColorSpan(Color.RED)
                }

                val start = resultadoText.indexOf(resultado)
                val end = start + resultado.length

                spannable.setSpan(colorSpan, start, end,
Spannable.SPAN_EXCLUSIVE_EXCLUSIVE)

                resultadoTextView.text = spannable

                resultadoTextView.visibility = View.VISIBLE
            } else {
                resultadoTextView.text = "Por favor, ingrese todas las
notas correctamente."
                resultadoTextView.setTextColor(Color.BLACK)
                resultadoTextView.visibility = View.VISIBLE
            }
        }
    }
}
```

7. Una vez que este completado el código, realice el proceso de compilación e instalación de la aplicación:



8. Una vez el proceso haya terminado la aplicación deberá verse de la siguiente manera:



Listo ahora solo agrega los datos de las notas y das clic en el botón calcular.

PRÁCTICA A REALIZAR

NOMBRE: Calculadora de notas

Indicación: Haciendo uso del ejemplo de la práctica y lo visto en clases, se requiere que desarrolle la siguiente aplicación:

Crear una aplicación que permita calcular las notas de un estudiante en **tres cómputos**, determinar si aprueba o no, y mostrar los resultados finales en una pantalla resumen.

Resultados esperados:

1. Diseño de interfaz amigable y apegada a los requerimientos.
2. Para la Activity de notas por cómputo se requiere:
    a. Solicita el **nombre del estudiante**.
    b. Debe permitir ingresar la nota mínima para aprobar.
    c. Establecer si son las notas del cómputo 1, 2, y 3 estas deben mantenerse en memoria de tal forma de poder generar los resultados por cómputo.
    d. Ejemplo: si selecciona Computo 1, y al poner las 3 notas, deberá calcular la nota promedio del cómputo 1 y así sucesivamente para los demás cómputos.
3. Se debe crear una Activity de Resultados Finales:

a) Debe mostrar el nombre del estudiante que se le está generando y un botón que diga "Ver Resultados Finales".
b) Botón "Calcular" para mostrar los resultados finales según siguiente detalle:
   - ✓ Estado final (Aprobado o No Aprobado), si aprobó, mostrar el mensaje "¡Aprobado!" con un color verde, si no aprobó, mostrar el mensaje "No Aprobado" con un color rojo.
   - ✓ Si no aprobó, indicar cuánto le falta para alcanzar la nota mínima.
   - ✓ Botón "Salir" para regresar al inicio.
c) Elabore un documento en Word, que contenga:
   a. La portada (con todas sus generalidades)
   b. Capturas de pantalla de como quedó su aplicación. (Incluya las pantallas principales).
   c. Haga un repositorio en git, suba el proyecto y adjunte una captura del repositorio con el respectivo enlace.
   d. Convierta el documento de WORD a PDF y adjúntelo al buzón de tareas correspondiente.

FORMA DE ENTREGA: Se deberá enviar al buzón de tarea llamado **Fundamentos de Kotlin**, de manera individual. **(No importa si lo han trabajado en parejas o equipos, siempre cada uno deberá adjuntar su evidencia en el buzón).**
TIPO DE ENTREGA: Enviar un documento en **PDF o DOCX** al final de la semana antes de las **23:59.**

RUBRICA DE EVALUACIÓN

Indicación: a continuación, se establecen los criterios de evaluación para la actividad de la semana.

#	CRITERIO	PTS.
1	Entrega el proyecto en la fecha establecida.	2
2	Diseña la interfaz principal de acuerdo con el requerimiento.	1.5
3	Diseña la interfaz de los resultados finales según requerimiento.	1.0
4	Funciona la interfaz principal de acuerdo con el requerimiento.	1.5
5	Funciona la interfaz de resultado de notas finales de acuerdo con el requerimiento.	2
6	Sube el proyecto a git y anexa el enlace en el documento.	1
7	El documento está ordenado y sin errores de ortografía.	1