



# UNIVERSIDAD DE COLIMA

Universidad de Colima

Facultad de Telemática

Ingeniería de software

Programación de Móviles

## **Trabajo Independiente - Dibujar: óvalos en Android Studio con Kotlin**

Tercera parcial

Almno. Angel Isaac Bejarano Flores

5°D

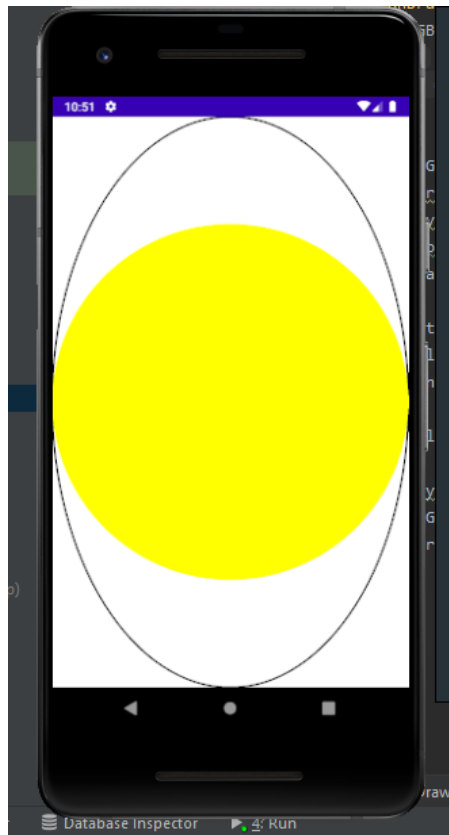
Mtro. Armando Román Gallardo

Martes 8 de diciembre de 2020. Manzanillo, Col.

## activity\_main.xml

```
<?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:id="@+id/layout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" />
```

## Interfaz



## MainActivity.kt

```
package com.example.dibujarovalos

import android.content.Context
import android.graphics.Canvas
import android.graphics.Paint
import android.graphics.RectF
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {
```

```

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

    val fondo = Lienzo(this)
    layout1.addView(fondo)
    supportActionBar!!.hide()
}

class Lienzo(context: Context) : View(context) {

    override fun onDraw(canvas: Canvas) {
        canvas.drawRGB(255, 255, 255)
        val ancho = getWidth()
        val alto = getHeight()

        val pincel1 = Paint()
        pincel1.setARGB(255, 0, 0, 0)
        pincel1.setStrokeWidth(5f)
        pincel1.setStyle(Paint.Style.STROKE)
        val rectangulo1 = RectF(0f, 0f, ancho.toFloat(), alto.toFloat())
        canvas.drawOval(rectangulo1, pincel1)

        val menor: Int
        if (ancho < alto)
            menor = ancho
        else
            menor = alto

        pincel1.setStyle(Paint.Style.FILL)
        pincel1.setARGB(255, 255, 255, 0)
        canvas.drawCircle((ancho / 2).toFloat(), (alto / 2).toFloat(),
(menor / 2).toFloat(), pincel1)
    }
}

```