



Universidad Tecnológica de Nezahualcóyotl

Carrera: Ing. Desarrollo y Gestión de Software.

Asignatura: Desarrollo Móvil para Dispositivos Inteligentes.

Profesor: Ricardo Armando Machorro Reyes.

Trabajo: Programas vistos en clase del segundo parcial Kotlin.

Alumno: Márquez Sánchez Eder Germán.

Matricula: 221271016.

Grupo: ITIC – 901M.

Periodo: Septiembre / diciembre 2022 – 3.

Fecha de Entrega: Domingo 30 de octubre del 2022.

OBJETOS

Principal.kt

```
package Objetos

import java.time.LocalDate
import java.time.Period
import java.time.format.DateTimeFormatter

class Alumno (var nombre: String, var fechaEnTexto: String){
    //constructor primario
    constructor():this("", "")

    //constructor secundario
    constructor(nombre: String):this("", ""){
        this.nombre = nombre
    }

    fun muestra(){
        //println("$nombre = $fechaEnTexto - ${obtenerEdad()} years")
        println("$nombre - ${if (fechaEnTexto.isNullOrEmpty()) "Sin Edad"
else obtenerEdad()} years")
    }

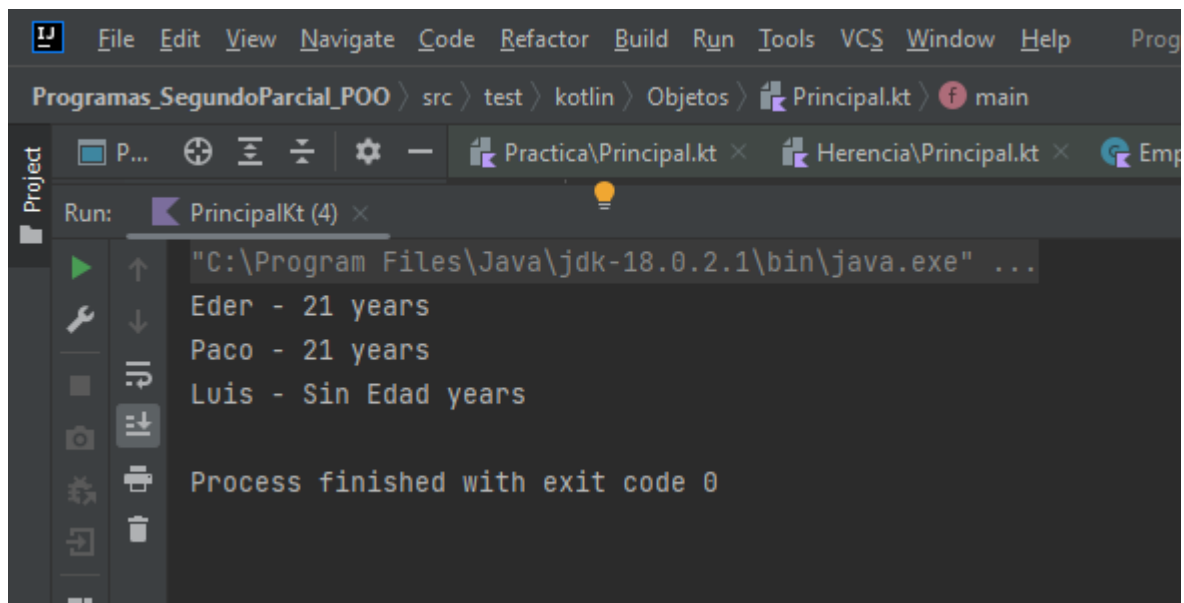
    fun obtenerEdad(): Int{
        val fechaDate = LocalDate.parse(fechaEnTexto,
DateTimeFormatter.ofPattern("dd/MM/yyyy"))
        return Period.between(fechaDate, LocalDate.now()).years
    }
}

fun main(){
    //Objeto 1
    val alumno1 = Alumno("Eder", "09/06/2001")
    alumno1.muestra()

    //Objeto2
    val alumno2 = Alumno()
    alumno2.nombre = "Paco"
    alumno2.fechaEnTexto = "19/05/2001"
    alumno2.muestra()

    //Objeto3
    val alumno3 = Alumno("Luis")
    alumno3.muestra()
}
```

Salida por consola.



HERENCIA

Administrador.kt

```
package Herencia

class Administrador(nombre: String, apellido: String, var noOficina: Int)
: Empleado(nombre, apellido) {
    override fun muestraTrabajo(): String{
        return super.muestraTrabajo() + " Planear reunion"
    }
}
```

Empleado.kt

```
package Herencia

open class Empleado(val nombre: String, val apellido: String) {
    var salario: Float = 0.0F
    set(value){
        field = value
    }
    get() = field - impuesto
    var impuesto: Float = 10.0F
    get() = 1 - (field * 0.1F)

    /*fun obtenerNombre(): String{
        return "$nombre $apellido"
    }*/

    open fun obtenerNombre() = "$nombre $apellido"

    open fun muestraTrabajo(): String{
        return "Capturando datos"
    }
}
```

Profesor.kt

```
package Herencia

class Profesor(nombre: String, apellido: String, var estudiantes: Short)
: Empleado(nombre, apellido) {
    override fun muestraTrabajo(): String{
        return super.muestraTrabajo() + " prepara la clase"
    }
}
```

Principal.kt

```
package Herencia

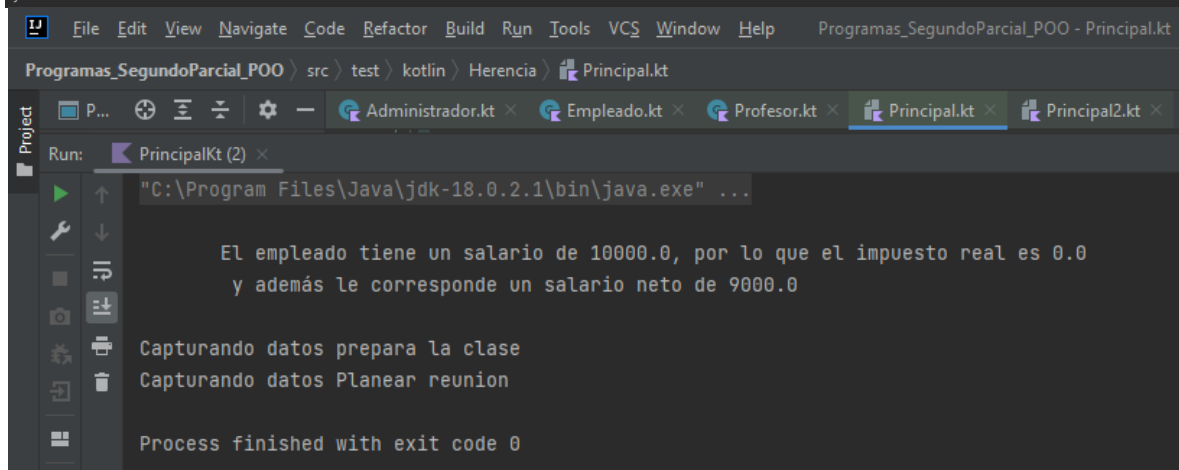
fun main() {
    val empleado = Empleado("Hugo", "Hernández")
    empleado.salarior = 10000.0F
    var salarioBruto = 10000.0F

    var salarioNeto = empleado.salarior * 0.9
    println("""
        El empleado tiene un salario de $salarioBruto, por lo que el
    impuesto real es ${empleado.impuesto}
        y además le corresponde un salario neto de $salarioNeto
    """)
    empleado.muestraTrabajo()

    val profesor = Profesor("Juanito", "Banana", 34)
    println(profesor.muestraTrabajo())

    val administrador = Administrador("Juanito", "Alvarado", 12)
    println(administrador.muestraTrabajo())
}


```



Principal2.kt

```
package Herencia

fun main() {
    val empleado1 = Empleado("Juan", "Alvarado")
    println("Nombre del empleado: ${empleado1.obtenerNombre()}")
    println("Actividad: ${empleado1.muestraTrabajo()}")
    empleado1.salarior = 10000.0F
    var salarioBruto = 10000.0F
    println("""
        Salario Bruto: $salarioBruto
        Cuando el impuesto es ${empleado1.impuesto} su salario neto es de
    ${'$'}${empleado1.salarior} pesos
    """).trimIndent()
}


```

```

println(" ==== GET & SET ==== ")
empleado1.impuesto = 20F
println("Cuando el impuesto es ${empleado1.impuesto} su salario es de
${'$'}${empleado1.salario} pesos")

println(" ===== HERENCIA ===== ")
val profesor = Profesor("Alex", "Castillo", 45)
val admin = Administrador("Gerardo", "Luna", 12)
println("===== EMPLEADO =====")
println(empleado1.obtenerNombre())
println("===== PROFESOR =====")
println(profesor.obtenerNombre())
println("===== ADMINISTRADOR =====")
println(admin.obtenerNombre())
}

```

The screenshot shows an IDE window titled "Programas_SegundoParcial_POO - Principal2.kt". The editor displays the Kotlin code from the previous block. Below the editor, the "Run" tab is active, showing the execution output. The output includes the program's execution path, user input for employee details, and the printed results of the code, including the "GET & SET" and "HERENCIA" sections.

```

Run: Principal2Kt x
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...
Nombre del empleado: Juan Alvarado
Actividad: Capturando datos
Salario Bruto: 10000.0
Cuando el impuesto es 0.0 su salario neto es de $10000.0 pesos
==== GET & SET ====
Cuando el impuesto es -1.0 su salario es de $10001.0 pesos
===== HERENCIA =====
===== EMPLEADO =====
Juan Alvarado
===== PROFESOR =====
Alex Castillo
===== ADMINISTRADOR =====
Gerardo Luna

Process finished with exit code 0

```


HERENCIA ABSTRACTA

Figura.kt

```
package Herencia_abstracta

abstract class Figura(var identificador: String){
    abstract fun obtenerArea(): Double
    abstract fun obtenerPerimetro(): Double
    abstract fun dibujar()
    protected fun centimetrosAPulgadas(valor: Double): Double{
        return (1 / 2.54) * valor
    }
}
```

Rectángulo.kt

```
package Herencia_abstracta

import kotlin.math.roundToInt

class Rectangulo(identificador: String, var base: Double, var altura: Double) : Figura(identificador) {
    override fun obtenerArea(): Double {
        return base * altura
    }

    override fun obtenerPerimetro(): Double {
        return (2 * base) + (2 * altura)
    }

    override fun dibujar() {
        for (i in 1..altura.roundToInt()){
            println(" * ".repeat(base.toInt()))
        }
    }

    fun obtenerAreaEnPulgadas(): Double{
        return centimetrosAPulgadas(obtenerArea())
    }

    /*fun obtenerPerimetroAPulgadas(): Double{
        val perimetro = obtenerPerimetro()
        val perimetroPulgada = centimetrosAPulgadas(perimetro)
        return perimetroPulgada
    }*/

    fun obtenerPerimetroAPulgadas() =
        centimetrosAPulgadas(obtenerPerimetro())
}
```

Principal.kt

```
package Herencia_abstracta

fun main(){
    val rectangulo = Rectangulo("Rectangulo", 10.0, 5.0)
    println(" ===== Clases Abstractas ===== ")
    println("")
}
```

```

    Area y Perimetro en Centimentros
    El area del rectangulo es: ${rectangulo.obtenerArea()}
    El perimetro del rectangulo es: ${rectangulo.obtenerPerimetro()}

    Area y Perimetro en Pulgadas
    El area del rectangulo es: ${rectangulo.obtenerAreaEnPulgadas()}
    El perimetro del rectangulo es:
    ${rectangulo.obtenerPerimetroAPulgadas()}
    "")
    println("La figura del rectangulo es:")
    rectangulo.dibujar()
}

```

Salida por consola.

```

"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...
===== Clases Abstractas =====

Area y Perimetro en Centimentros
El area del rectangulo es: 50.0
El perimetro del rectangulo es: 30.0

Area y Perimetro en Pulgadas
El area del rectangulo es: 19.685039370078737
El perimetro del rectangulo es: 11.811023622047243

La figura del rectangulo es:
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

Process finished with exit code 0

```

Interfaces.

Figura.kt

```
package Herencia_abstracta.Interfaces

interface Figura {
    var identificador: String
    fun obtenerArea(): Double
    fun obtenerPerimetro(): Double
    fun dibujar()
    fun centimetrosAPulgadas(valor: Double): Double{
        return (1 / 2.54) * valor
    }
}
```

Rectángulo.kt

```
package Herencia_abstracta.Interfaces

import kotlin.math.roundToInt

class Rectangulo(override var identificador: String, var base: Double,
var altura: Double) : Figura{
    override fun obtenerArea(): Double {
        return base * altura
    }

    override fun obtenerPerimetro(): Double {
        return (2 * base) + (2 * altura)
    }

    override fun dibujar() {
        for (i in 1..altura.roundToInt()){
            println(" * ".repeat(base.toInt()))
        }
    }

    fun obtenerAreaEnPulgadas(): Double{
        return centimetrosAPulgadas(obtenerArea())
    }

    fun obtenerPerimetroAPulgadas() =
        centimetrosAPulgadas(obtenerPerimetro())
}
```

Principal.kt

```
package Herencia_abstracta.Interfaces

fun main(){
    val rectangulo = Rectangulo("Rectangulo", 10.0, 5.0)
    println("==== Interfaces ==== ")
    println("""
        Area y Perimetro en Centimetros
        El area del rectangulo es: ${rectangulo.obtenerArea()}
    """)
}
```

```

        El perimetro del rectangulo es: ${rectangulo.obtenerPerimetro()}

        Area y Perimetro en Pulgadas
        El area del rectangulo es: ${rectangulo.obtenerAreaEnPulgadas()}
        El perimetro del rectangulo es:
        ${rectangulo.obtenerPerimetroAPulgadas()}
        """)
        println("La figura del rectangulo es:")
        rectangulo.dibujar()
    }
}

```

Salida por consola.

```

"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...
===== Interfaces =====

Area y Perimetro en Centimetros
El area del rectangulo es: 50.0
El perimetro del rectangulo es: 30.0

Area y Perimetro en Pulgadas
El area del rectangulo es: 19.685039370078737
El perimetro del rectangulo es: 11.811023622047243

La figura del rectangulo es:
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

Process finished with exit code 0

```

ANDROID STUDIO

Programa Saludo con findViewById.

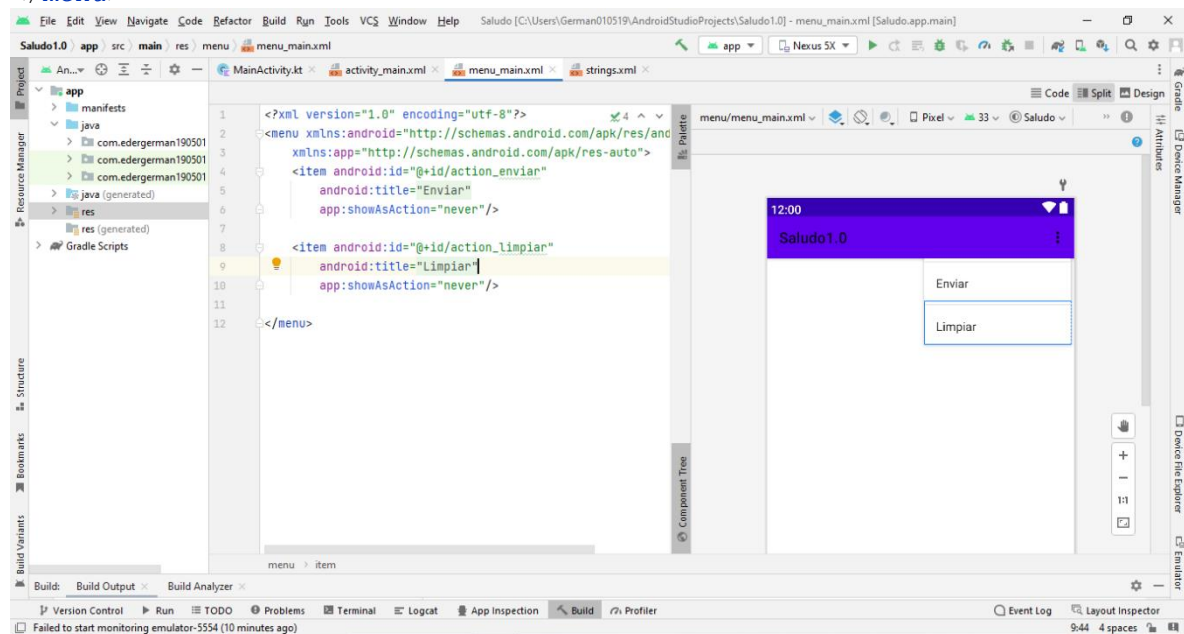
Strings.xml.

```
<resources>
    <string name="app_name">Saludo1.0</string>
    <string name="titulo">Android Studio Saludo findViewById</string>
    <string name="tvNombre">Nombre</string>
    <string name="btnEnviar">Enviar</string>
    <string name="itemEnviar">Enviar</string>
    <string name="itemLimpiar">Limpiar</string>
</resources>
```

Menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item android:id="@+id/action_enviar"
        android:title="@string/itemEnviar"
        app:showAsAction="never"/>

    <item android:id="@+id/action_limpiar"
        android:title="@string/itemLimpiar"
        app:showAsAction="never"/>
</menu>
```



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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tvTitulo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="32dp"
        android:layout_marginEnd="32dp"
        android:text="@string/titulo"
        android:textColor="@color/black"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tvNombre"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="32dp"
        android:text="@string/tvNombre"
        android:textColor="@color/black"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvTitulo" />

    <EditText
        android:id="@+id/etNombre"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="32dp"
        android:autofillHints="name"
        android:ems="30"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvNombre"

    tools:ignore="LabelFor,TouchTargetSizeCheck,SpeakableTextPresentCheck" />
```



```

<TextView
    android:id="@+id/tvSalida"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="32dp"
    android:textStyle="bold"
    android:textColor="@color/black"
    android:textSize="20sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etNombre"/>

```

```

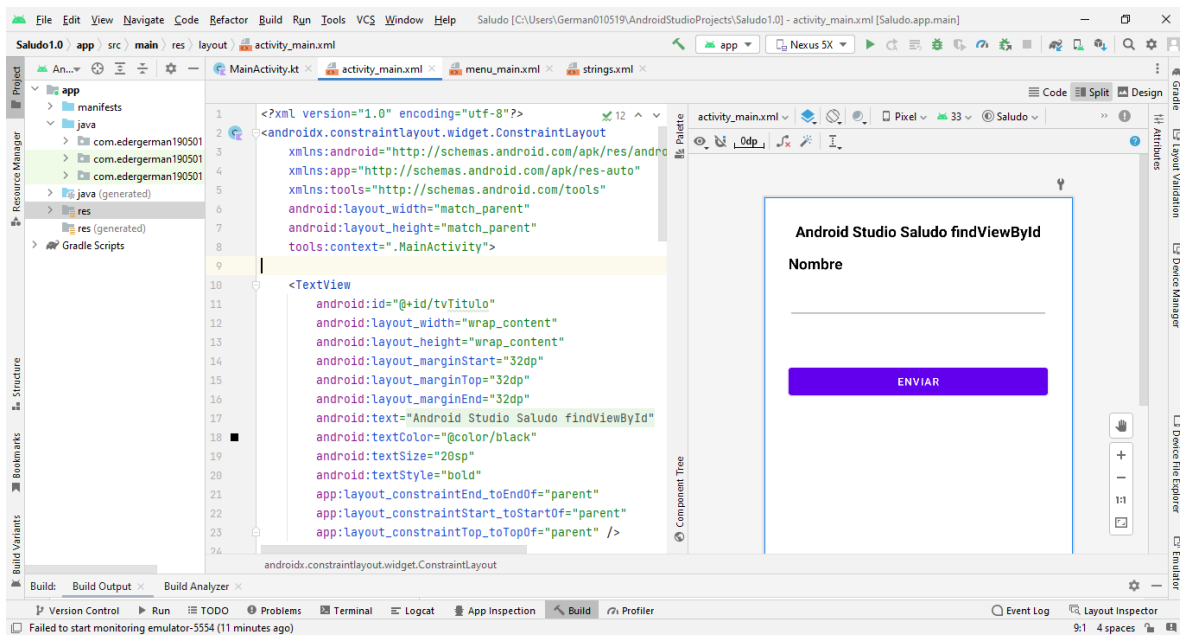
<Button
    android:id="@+id/btnEnviar"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="32dp"
    android:text="@string/btnEnviar"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvSalida"/>

```

```

</androidx.constraintlayout.widget.ConstraintLayout>

```



MainActivity.kt

```
package com.edergerman190501

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast

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

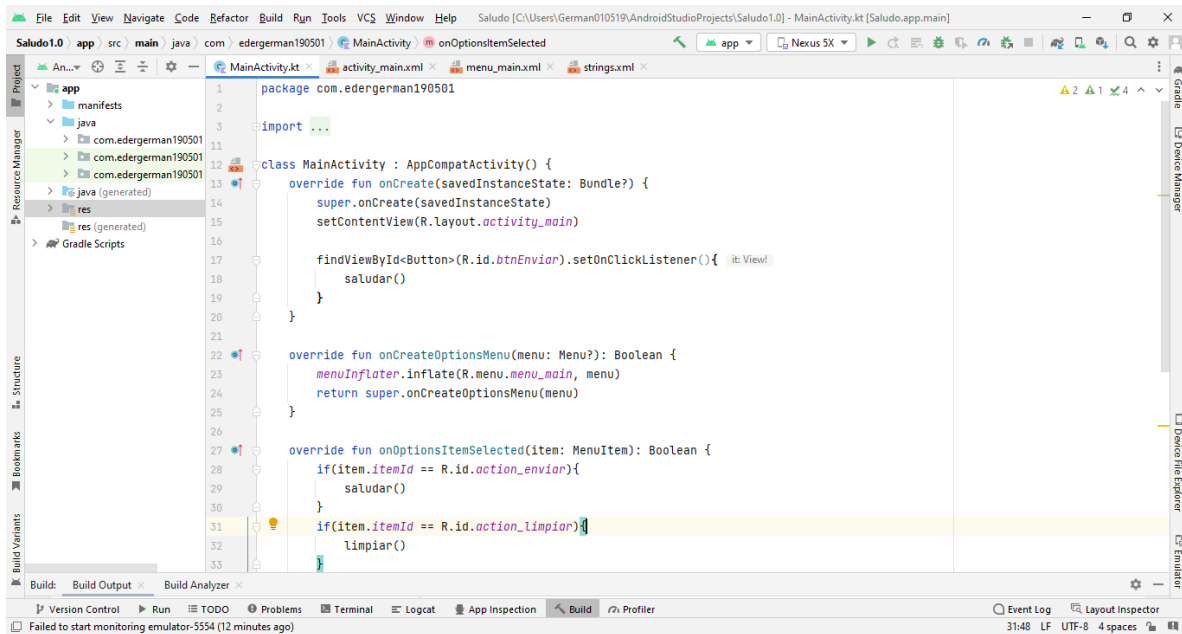
        findViewById<Button>(R.id.btnEnviar).setOnClickListener() {
            saludar()
        }

        override fun onCreateOptionsMenu(menu: Menu?): Boolean {
            menuInflater.inflate(R.menu.menu_main, menu)
            return super.onCreateOptionsMenu(menu)
        }

        override fun onOptionsItemSelected(item: MenuItem): Boolean {
            if (item.itemId == R.id.action_enviar) {
                saludar()
            }
            if (item.itemId == R.id.action_limpiar) {
                limpiar()
            }
            return super.onOptionsItemSelected(item)
        }

        private fun saludar() {
            val name =
                findViewById<EditText>(R.id.etNombre).text.toString().trim()
                findViewById<TextView>(R.id.tvSalida).text = "Hola $name"
                Toast.makeText(this, "Hola $name", Toast.LENGTH_LONG).show()
        }

        private fun limpiar() {
            findViewById<EditText>(R.id.etNombre).text.clear()
            findViewById<TextView>(R.id.tvSalida).text = null
        }
    }
}
```



Programa Saludo con viewBiding.

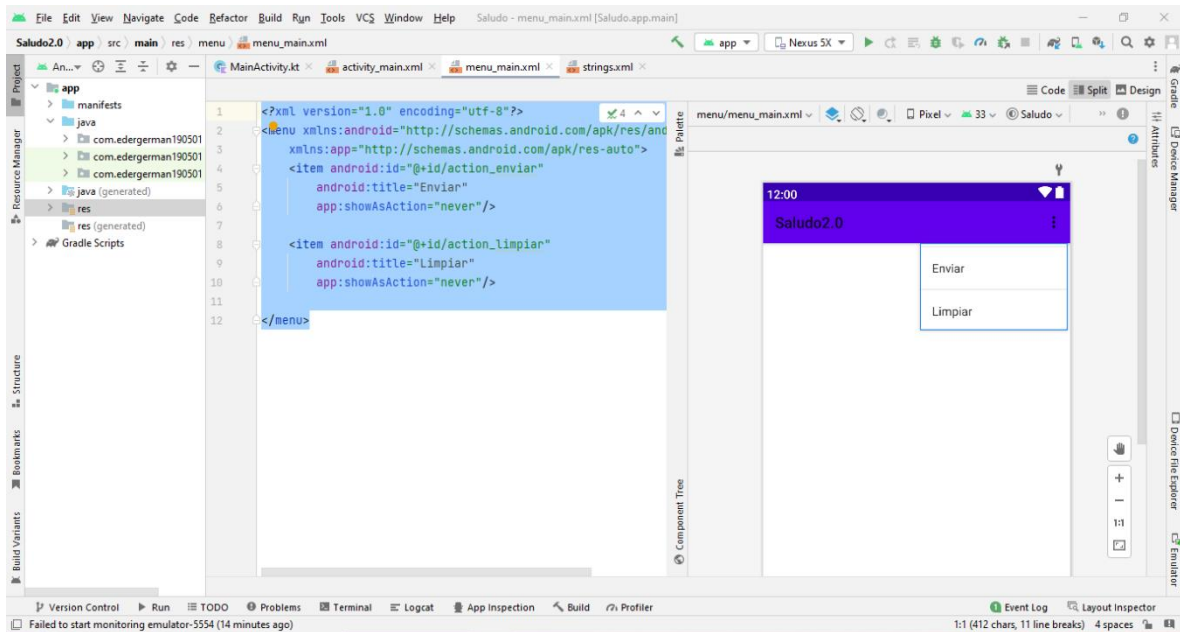
Strings.xml.

```
<resources>
    <string name="app_name">Saludo2.0</string>
    <string name="titulo">Android Studio Saludo viewBading</string>
    <string name="tvNombre">Nombre</string>
    <string name="btnEnviar">Enviar</string>
    <string name="itemEnviar">Enviar</string>
    <string name="itemLimpiar">Limpiar</string>
</resources>
```

Menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item android:id="@+id/action_enviar"
        android:title="@string/itemEnviar"
        app:showAsAction="never"/>

    <item android:id="@+id/action_limpiar"
        android:title="@string/itemLimpiar"
        app:showAsAction="never"/>
</menu>
```



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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tvTitulo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="32dp"
        android:layout_marginEnd="32dp"
        android:text="@string/titulo"
        android:textColor="@color/black"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tvNombre"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="32dp"
        android:text="@string/tvNombre"
```

```

        android:textColor="@color/black"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvTitulo" />

<EditText
    android:id="@+id/etNombre"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="32dp"
    android:autofillHints="name"
    android:ems="30"
    android:inputType="textPersonName"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvNombre"

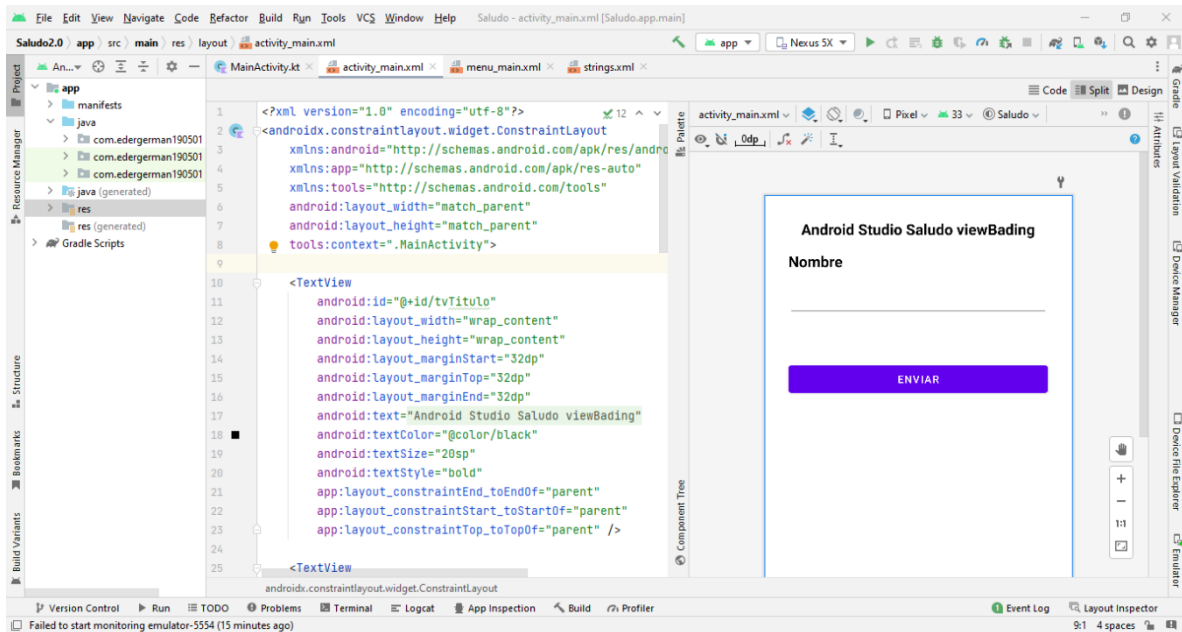
tools:ignore="LabelFor,TouchTargetSizeCheck,SpeakableTextPresentCheck" />

<TextView
    android:id="@+id/tvSalida"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="32dp"
    android:textStyle="bold"
    android:textColor="@color/black"
    android:textSize="20sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etNombre"/>

<Button
    android:id="@+id/btnEnviar"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="32dp"
    android:text="@string/btnEnviar"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvSalida"/>

</androidx.constraintlayout.widget.ConstraintLayout>

```



MainActivity.kt

```
package com.edergerman190501

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.widget.Toast
import com.edergerman190501.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.btnEnviar.setOnClickListener {
            saludar()
        }
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.menu_main, menu)
        return super.onCreateOptionsMenu(menu)
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        if (item.itemId == R.id.action_enviar) {
            saludar()
        }
        if (item.itemId == R.id.action_limpiar) {

```

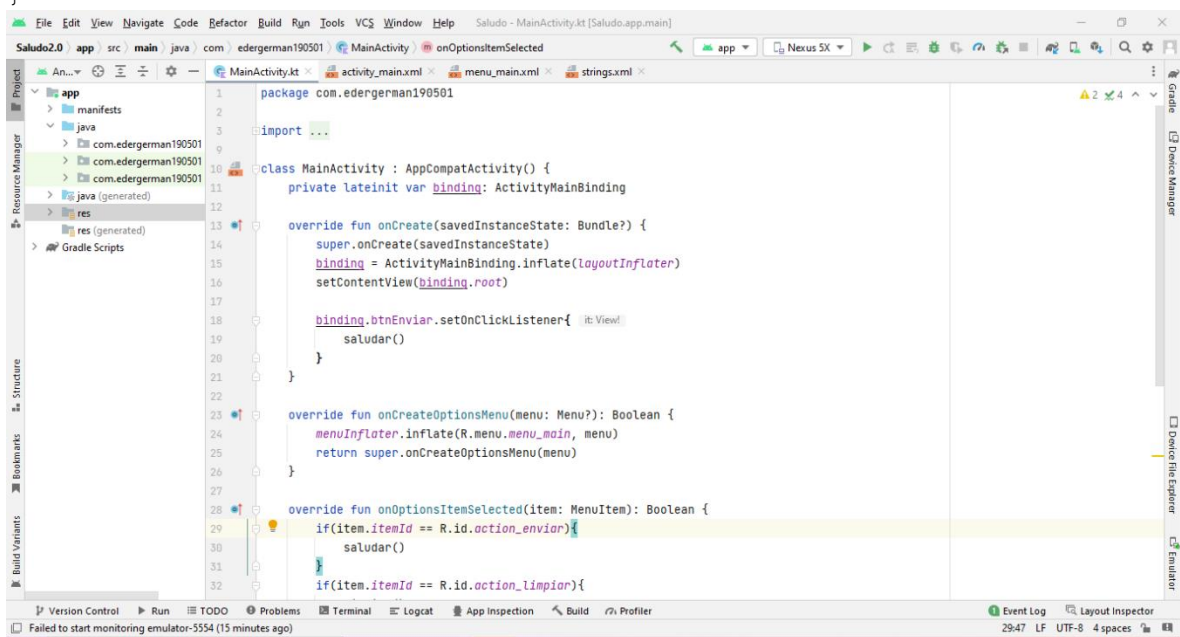
```

        limpiar()
    }
    return super.onOptionsItemSelected(item)
}

private fun saludar() {
    val name = binding.etNombre.text.toString().trim()
    binding.tvSalida.text = "Hola $name"
    Toast.makeText(this, "Hola $name", Toast.LENGTH_LONG).show()
}

private fun limpiar() {
    binding.etNombre.text.clear()
    binding.tvSalida.text = null
}
}

```



Programa División.

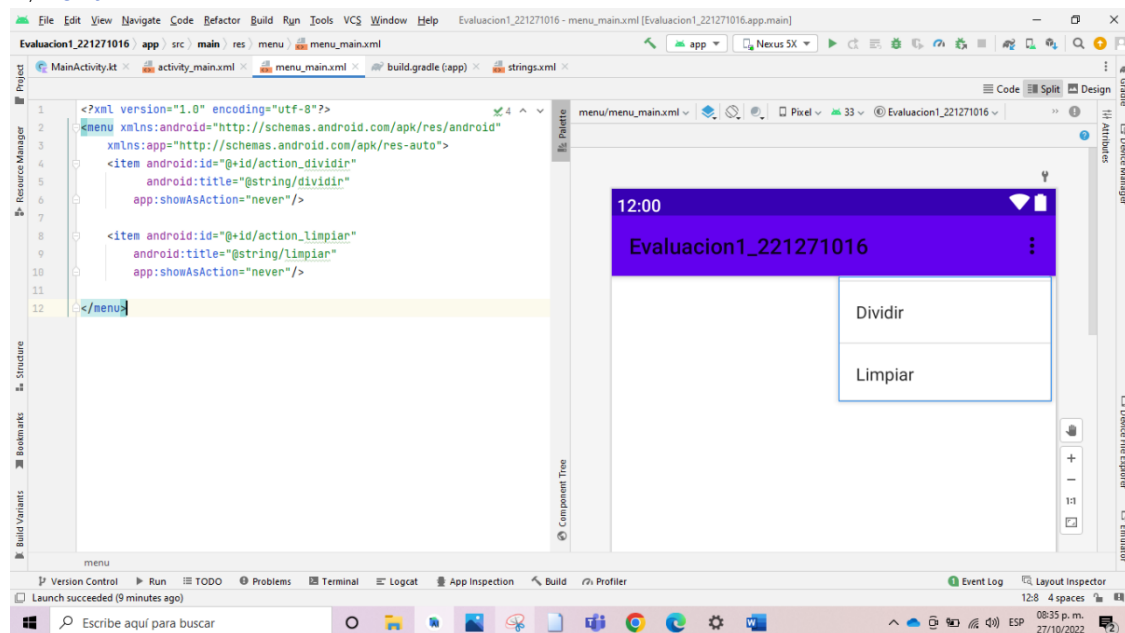
Strings.xml.

```
<resources>
    <string name="app_name">Evaluacion1_221271016</string>
    <string name="dividir">Dividir</string>
    <string name="limpiar">Limpiar</string>
    <string name="ecuacion">Mi ecuación</string>
    <string name="txtA">a</string>
    <string name="txtB">b</string>
    <string name="btnDividir">Dividir</string>
</resources>
```

Menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item android:id="@+id/action_dividir"
        android:title="@string/dividir"
        app:showAsAction="never"/>

    <item android:id="@+id/action_limpiar"
        android:title="@string/limpiar"
        app:showAsAction="never"/>
</menu>
```



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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/lblEcuacion"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:layout_marginHorizontal="16dp"
        android:text="@string/ecuacion"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="@color/black"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
    />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:layout_marginHorizontal="16dp"
        app:srcCompat="@drawable/img"
        android:contentDescription="@string/ecuacion"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/lblEcuacion" />

    <TextView
        android:id="@+id/lblA"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:layout_marginHorizontal="16dp"
        android:text="@string/txtA"
        android:textSize="30sp"
        android:textStyle="bold"
        android:autoFillHints="@string/txtA"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView"
    />

    <EditText
        android:id="@+id/txtA"
```

```

        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginHorizontal="16dp"
        android:layout_marginTop="16dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:autofillHints="name"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/lblA"
        tools:ignore="LabelFor"/>

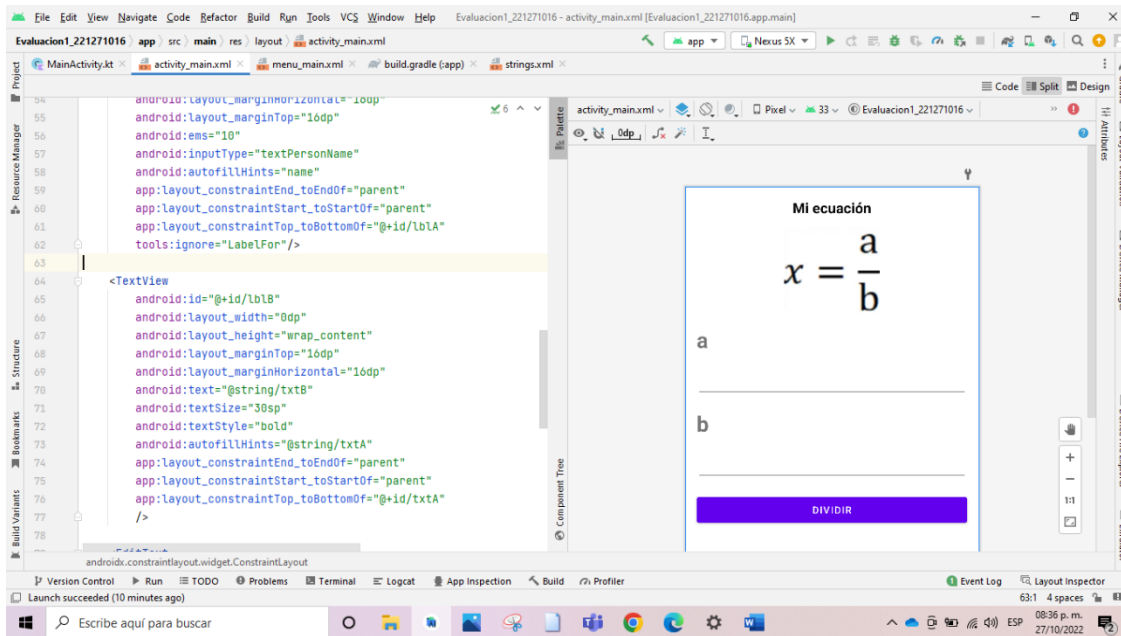
<TextView
    android:id="@+id/lblB"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:layout_marginHorizontal="16dp"
    android:text="@string/txtB"
    android:textSize="30sp"
    android:textStyle="bold"
    android:autofillHints="@string/txtA"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/txtA"
    />

<EditText
    android:id="@+id/txtB"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="16dp"
    android:layout_marginTop="16dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:autofillHints="name"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/lblB"
    tools:ignore="LabelFor"/>

<Button
    android:id="@+id/btnDividir"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:layout_marginHorizontal="16dp"
    android:text="@string/btnDividir"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/txtB"/>

</androidx.constraintlayout.widget.ConstraintLayout>

```



MainActivity.kt

```
package com.example.evaluacion1_221271016

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.widget.EditText
import android.widget.Toast
import com.example.evaluacion1_221271016.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.btnDividir.setOnClickListener {
            calcular()
        }
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.menu_main, menu)
        return super.onCreateOptionsMenu(menu)
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        if (item.itemId == R.id.action_dividir) {
            calcular()
            return true
        }
    }
}
```

```

        if(item.itemId == R.id.action_limpiar){
            limpiar()
            return true
        }
        return super.onOptionsItemSelected(item)
    }

    private fun calcular() {
        var ok = true
        var a = 0.0

        try {
            a = leeDouble(binding.txtA)
        } catch (e: NumberFormatException){
            ok = false
        }

        var b = 0.0

        try {
            b = leeDouble(binding.txtB)
        } catch (e: NumberFormatException){
            ok = false
        }

        if(ok){
            val x = a / b
            if(b != 0.0){
                mensaje("x = $x")
            } else{
                mensaje("No se permite la division entre cero")
            }
        }
    }

    private fun mensaje(mensaje: String) {
        Toast.makeText(this, mensaje, Toast.LENGTH_LONG).show()
    }

    private fun leeDouble(texto: EditText): Double {
        return try{
            texto.text.toString().toDouble()
        } catch (e: NumberFormatException){
            texto.error = "Valor Incorrecto"
            throw e
        }
    }

    private fun limpiar() {
        binding.txtA.setText("")
        binding.txtA.error=null
        binding.txtB.setText("")
        binding.txtB.error=null
    }
}

```

