

Universidad Don Bosco

Desarrollo de software para móviles DSM 04L



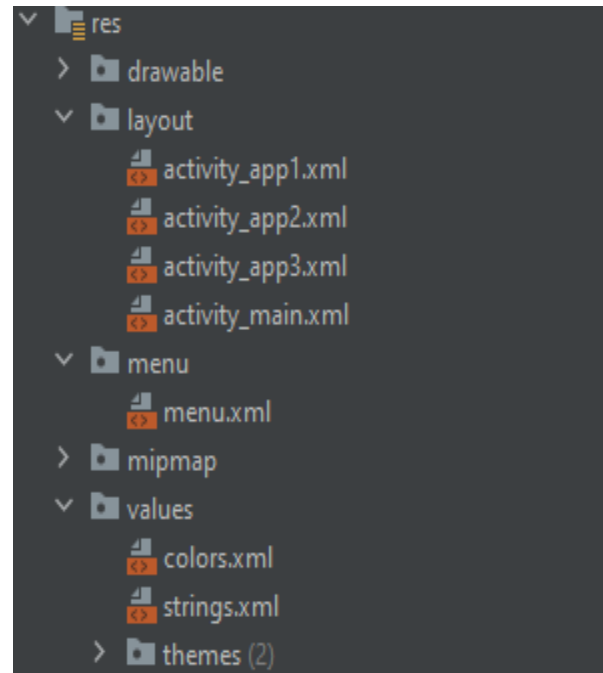
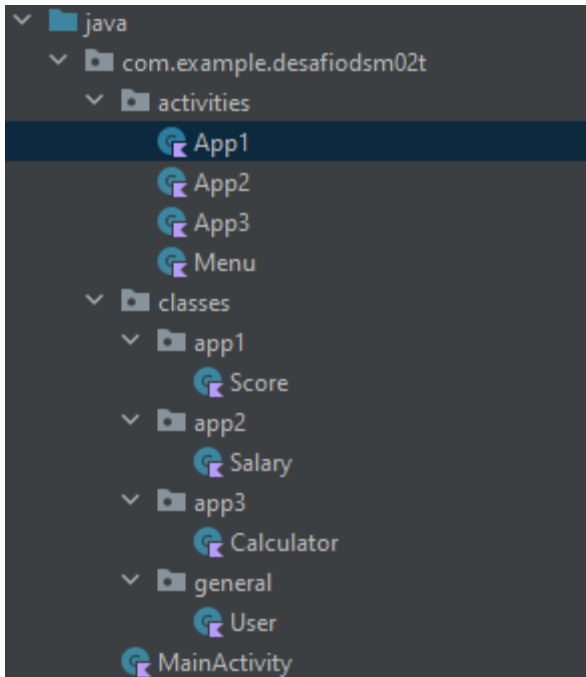
Profesor: Alexander Alberto Sigüenza Campos

Alumno: Ruben Oswaldo Escalante Amaya

Fecha: 26/02/2023

Problema 1: Promedio de notas Completo al 100%

Para resolver los problemas se uso una estructura de carpetas simple, una carpeta donde están las actividades, otra para las clases que se ocuparan los ejercicios.



Para el problema 1 ocupamos la clase Score que recibe en su constructor un array de tipo Long que cuenta con 2 metodos, average() que nos saca el promedio y validation() que nos regresa si el alumno aprobó o reprobó.

```
class Score(private val scores: LongArray) {  
  
    fun average(): Long{  
        return scores.average().toLong()  
    }  
  
    fun validation(): String{  
        val validation = if (average() < 6){  
            "Reprobado"  
        } else{  
            "Aprobado"  
        }  
        return validation  
    }  
}
```

También se usó la clase User donde estará toda la lógica para tratar el nombre del usuario.

```
class User(private val name_: String,private val lastname_: String) {

    fun name(): String{
        return name_
    }

    fun lastName(): String{
        return lastname_
    }

    fun fullName(): String{
        return "${name()} ${lastName()}"
    }
}
```

La actividad para este problema es App1 donde mayormente vinculamos los elementos de la vista y la actividad e instanciamos las clases.

```
class App1: Menu() {
    "RubenEscalante"
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_app1)

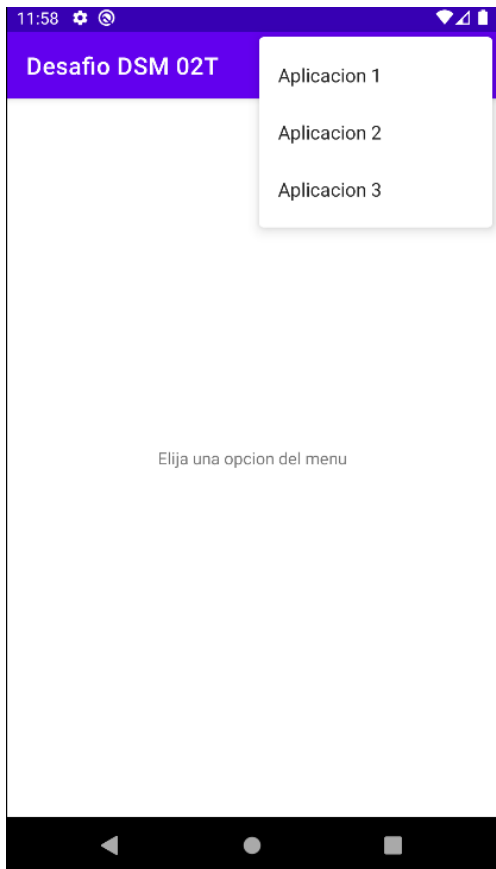
        val button = findViewById<Button>(R.id.btnApp1)

        button.setOnClickListener { it: View!
            val name = findViewById<EditText>(R.id.txtName)
            val lastname = findViewById<EditText>(R.id.txtLastName)
            val result = findViewById<TextView>(R.id.tvResult)
            var arrayScr = LongArray( size: 5)
            arrayScr[0] = findViewById<EditText>(R.id.txtScore1).text.toString().toLong()
            arrayScr[1] = findViewById<EditText>(R.id.txtScore2).text.toString().toLong()
            arrayScr[2] = findViewById<EditText>(R.id.txtScore3).text.toString().toLong()
            arrayScr[3] = findViewById<EditText>(R.id.txtScore4).text.toString().toLong()
            arrayScr[4] = findViewById<EditText>(R.id.txtScore5).text.toString().toLong()

            val userApp1 = User(name.text.toString(), lastname.text.toString())
            val scoreAvg = com.example.desafiodsm02t.classes.app1.Score(arrayScr)

            result.text = "Resultado para: ${userApp1.fullName()} \n Nota promedio: ${scoreAvg.average()} El alumno ha ${scoreAvg.validation()}"
        }
    }
}
```

Y las vistas son 7 edit text, 2 para el nombre y apellido y 5 para las notas. Un botón para calcular el promedio.



Desafio DSM 02T

| Ruben | Escalante |
|--------|-----------|
| Nota 1 | 7 |
| Nota 2 | 6 |
| Nota 3 | 8 |
| Nota 4 | 3 |
| Nota 5 | 7 |

CALCULAR PROMEDIO

Resultado para: Ruben Escalante
Nota promedio: 6 El alumno ha Aprobado

Problema 2: Salario Neto Completo al 100%

Para este problema ocupamos la clase Salary con una simple implementación

```
class Salary(private val salary_: Double) {  
    fun getISSS():Double{  
        return salary_ * 0.03  
    }  
    fun getAFP():Double{  
        return salary_ * 0.04  
    }  
    fun getRent():Double{  
        return salary_ * 0.05  
    }  
    fun getNetSalary():Double{  
        return (salary_ - getISSS() - getAFP() - getRent())  
    }  
}
```

Su actividad es App2

```
class App2: Menu() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_app2)  
  
        val button = findViewById<Button>(R.id.btnApp2)  
        val result = findViewById<TextView>(R.id.tvResultApp2)  
  
        button.setOnClickListener{ it: View! -> {  
            val name = findViewById<EditText>(R.id.txtNameApp2).text.toString()  
            val salary = findViewById<EditText>(R.id.txtSalaryApp2).text.toString().toDouble()  
            val netsalary = Salary(salary).getNetSalary()  
            result.text = "Su salario neto $name es $netsalary"  
        }}  
    }  
}
```

Desafio DSM 02T



Nombre

Ruben Escalante

Salario base

1200

CALCULAR SALARIO NETO

Su salario neto Ruben Escalante es 1056.0

1

2

3

-

4

5

6

⌋

7

8

9

⌫

,

0

.



Problema 3: Calculadora Básica completo 100%

Para este problema se utilizó un spinner para elegir las operaciones



Para programar el spinner se utilizó la siguiente lógica:

```
fun populateSpinner(): String{
    var spinnerSelected = ""
    val operaciones = arrayOf("+", "-", "x", "/")
    val spinner: Spinner = findViewById(R.id.spinner)
    val arrayAdapter =
        ArrayAdapter(context = this, android.R.layout.simple_spinner_item, operaciones)
    spinner.adapter = arrayAdapter
    spinner.onItemSelectedListener = object : AdapterView.OnItemSelectedListener{
        override fun onItemSelected(p0: AdapterView<*>?, p1: View?, p2: Int, p3: Long) {
            spinnerSelected = operaciones[p2]
        }
        override fun onNothingSelected(p0: AdapterView<*>?) {}
    }
    return spinnerSelected
}
```

Y luego se le asigna el valor que retorna esta función a una variable en onCreate de la actividad de la app3.

```
class App3: Menu() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_app3)

        var spinnerSelected = populateSpinner()
        val button = findViewById<Button>(R.id.btnApp3)
        val resulttv = findViewById<TextView>(R.id.tvResultApp3)

        button.setOnClickListener { it: View!
            val num1 = findViewById<EditText>(R.id.txtnum1).text.toString().toDouble()
            val num2 = findViewById<EditText>(R.id.txtnum2).text.toString().toDouble()
            val result = Calculator(num1, num2, spinnerSelected).getResult()
            resulttv.text = "El resultado es ${result}"
        }
    }
}
```

La clase para este problema es Calculator que implementa toda la lógica de la calculadora:

```
class Calculator(val num1: Double, val num2: Double, val operation: String) {  
    fun sum(): Double {  
        return num1 + num2  
    }  
    fun resta(): Double {  
        return num1 - num2  
    }  
    fun mul(): Double {  
        return num1 * num2  
    }  
    fun divi(): Double {  
        return num1 / num2  
    }  
    fun getResult(): Double {  
        var result = 0.0  
        when (operation) {  
            "+" -> result = sum()  
            "-" -> result = resta()  
            "x" -> result = mul()  
            "/" -> result = divi()  
            else -> {  
                result = 0.0  
            }  
        }  
        return result  
    }  
}
```


Desafío DSM 02T

5 x 7

CALCULAR

El resultado es 35.0

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | - |
| 4 | 5 | 6 | = |
| 7 | 8 | 9 | ✕ |

Desafío DSM 02T

Desafío DSM 02T

5 + 7

CALCULAR

El resultado es 12.0

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | - |
| 4 | 5 | 6 | = |
| 7 | 8 | 9 | ✕ |

Desafío DSM 02T

5 - 7

CALCULAR

El resultado es -2.0

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | - |
| 4 | 5 | 6 | = |
| 7 | 8 | 9 | ✕ |
| , | 0 | . | ✓ |

5 / 7

CALCULAR

El resultado es 0.7142857142857143

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | - |
| 4 | 5 | 6 | = |
| 7 | 8 | 9 | ✕ |
| , | 0 | . | ✓ |

Para el Menu se creo una actividad aparte y por la cual las demás heredan de esta, de esta manera solo se crea la lógica del menú aquí y no se necesita ponerla en las demás actividades que comparte el menú.

```
open class Menu: AppCompatActivity() {
    @ RubenEscalante
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.menu, menu)
        return true
    }
    @ RubenEscalante
    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        // Handle item selection
        return when (item.itemId) {
            R.id.view_app1 -> {
                val intent = Intent( packageContext: this, App1::class.java)
                startActivity(intent)
                true
            }
            R.id.view_app2 -> {
                val intent = Intent( packageContext: this, App2::class.java)
                startActivity(intent)
                true
            }
            R.id.view_app3 -> {
                val intent = Intent( packageContext: this, App3::class.java)
                startActivity(intent)
                true
            }
            else -> super.onOptionsItemSelected(item)
        }
    }
}
```