Capítulo 2 - Fundamentos Kotlin y Compose

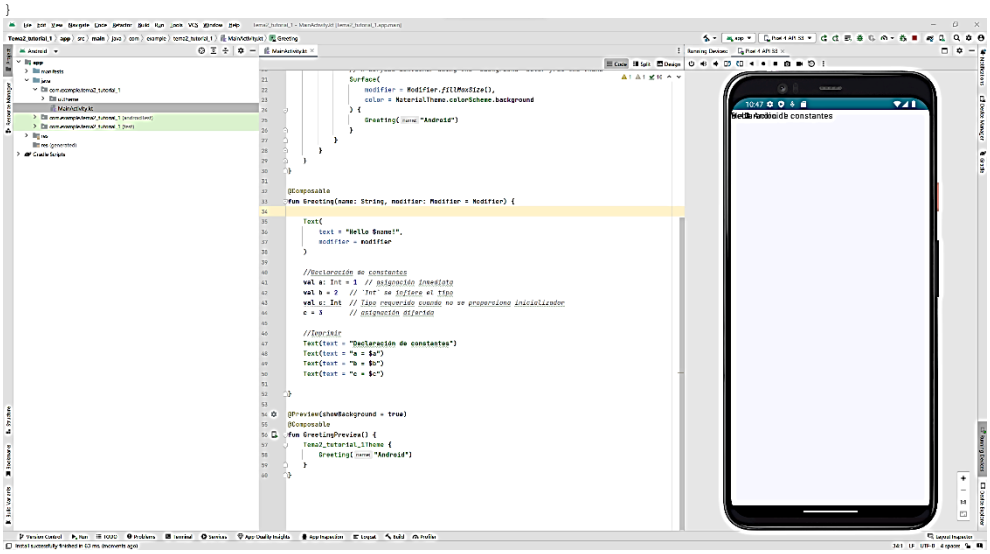
```
package com.example.tema2_tutorial_1

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_tutorial_1.ui.theme.Tema1_tutorial_1Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_tutorial_1Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Android")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Text(
        text = "Hello $name!",
        modifier = modifier
    )
}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_tutorial_1Theme {
        Greeting("Android")
    }
}
```

Constantes:
Son variables que solamente se les asigna una vez el valor y no podrá ser cambiado. Van precedidas de la palabra reservada **val**.

Ejemplo:

```
package com.example.tema2_tutorial_1

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_tutorial_1.ui.theme.Tema1_tutorial_1Theme
```

```
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_tutorial_1Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Android")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Text(
        text = "Hello $name!",
        modifier = modifier
    )

    //Declaración de constantes
    val a: Int = 1  // asignación inmediata
    val b = 2   // `Int` se infiere el tipo
    val c: Int  // Tipo requerido cuando no se proporciona inicializador
    c = 3       // asignación diferida

    //Imprimir
    Text(text = "Declaración de constantes")
    Text(text = "a = $a")
    Text(text = "b = $b")
    Text(text = "c = $c")

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_tutorial_1Theme {
        Greeting("Android")
    }
}
```



Ahora se realizará un estudio de cómo se estructura el programa:

```
package com.example.tema2_tutorial_1
```

```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_tutorial_1.ui.theme.Tema1_tutorial_1Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_tutorial_1Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Android")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column {
        Text(
            text = "Hello $name!",
            modifier = modifier
        )

        //Declaración de constantes
        val a: Int = 1  // asignación inmediata
        val b = 2   // `Int` se infiere el tipo
        val c: Int  // Tipo requerido cuando no se proporciona inicializador
        c = 3       // asignación diferida

        //Imprimir
        Text(text = "Declaración de constantes")
        Text(text = "a = $a")
        Text(text = "b = $b")
        Text(text = "c = $c")
    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_tutorial_1Theme {
        Greeting("Android")
    }
}
```

Código de **MainActivity.kt**:

```
package com.example.tema1_tutorial_1

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema1_tutorial_1.ui.theme.Tema1_tutorial_1Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
```

```
        setContent {
            Tema1_tutorial_1Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Android")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column {
        //líneas de código que influye la etiqueta Column
        Text(
            text = "Hello $name!",
            modifier = modifier
        )

        //Declaración máxima y mínima representación de entero según formato
        //Utilizando la palabra reservada var se declaran variables reasignables
        var MaxByte: Byte = Byte.MAX_VALUE
        var MinByte: Byte = Byte.MIN_VALUE

        var MaxShort: Short = Short.MAX_VALUE
        var MinShort: Short = Short.MIN_VALUE

        var MaxInt: Int = Int.MAX_VALUE
        var MinInt: Int = Int.MIN_VALUE

        var MaxLong: Long = Long.MAX_VALUE
        var MinLong: Long = Long.MIN_VALUE

        //Declaración de variables
        var unByte: Byte = 1
        var ByteSizeBits = Byte.SIZE_BITS
        var ByteSizeByte = Byte.SIZE_BYTES

        var unShort: Short = 1
        var ShortSizeBits = Short.SIZE_BITS
        var ShortSizeByte = Short.SIZE_BYTES

        var unInt: Int = 1
        var IntSizeBits = Int.SIZE_BITS
        var IntSizeByte = Int.SIZE_BYTES

        var unLong: Long = 1
        var LongSizeBits = Long.SIZE_BITS
        var LongSizeByte = Long.SIZE_BYTES


        //Mostrar por pantalla
        Text(text = "==================================")
        Text(text = "Mostrar de variables")
        Text(text = "==================================")
        Text(text = "unByte = $unByte")
        Text(text = "Longitud Byte en Bits = $ByteSizeBits")
        Text(text = "Longitud Byte en Bytes = $ByteSizeByte")
        Text(text ="Máximo valor de Byte: " + MaxByte)
        Text(text = "Mínimo valor de Byte: $MinByte")
        Text(text = "El tipo de la variable es: ${unByte::class.simpleName}")
        Text(text = "==================================")
        Text(text = "unShort = $unShort")
        Text(text = "Longitud Short en Bits = $ShortSizeBits")
        Text(text = "Longitud Short en Bytes = $ShortSizeByte")
        Text(text ="Máximo valor de Short: " + MaxShort)
        Text(text = "Mínimo valor de Short: $MinShort")
        Text(text = "El tipo de la variable es: ${unShort::class.simpleName}")
        Text(text = "==================================")
        Text(text = "unInt = $unInt")
        Text(text = "Longitud Int en Bits = $IntSizeBits")
        Text(text = "Longitud Int en Bytes = $IntSizeByte")
        Text(text ="Máximo valor de Int: " + MaxInt)
        Text(text = "Mínimo valor de Int: $MinInt")
        Text(text = "El tipo de la variable es: ${unInt::class.simpleName}")
        Text(text = "==================================")
        Text(text = "unLong = $unLong")
```

```
        Text(text = "Longitud Long en Bits = $LongSizeBits")
        Text(text = "Longitud Long en Bytes = $LongSizeByte")
        Text(text ="Máximo valor de Long: " + MaxLong)
        Text(text = "Mínimo valor de Long: $MinLong")
        Text(text = "El tipo de la variable es: ${unLong::class.simpleName}")
        Text(text = "==================================")

    }


}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_tutorial_1Theme {
        Greeting("Android")
    }
```

MainActivity.kt:

```
package com.example.tema2_tutorial_2

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_tutorial_2.ui.theme.Tema1_tutorial_2Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_tutorial_2Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Android")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Column {
        //líneas de código que influye la etiqueta Column
        Text(text = "Tipo $name!",color= Color.Red)


        //Declaración máxima y mínima representación de un real según formato
        var MaxFloat: Float = Float.MAX_VALUE
        var MinFloat: Float = Float.MIN_VALUE

        var MaxDouble: Double = Double.MAX_VALUE
        var MinDouble: Double = Double.MIN_VALUE

        //Declaración de variables
        //Para Float hay que poner una f
        //Con float se aprovecha mejor el espacio en memoria
        var unFloat: Float = 1.0f
        var FloatSizeBits = Float.SIZE_BITS
        var FloatSizeByte = Float.SIZE_BYTES

        var unDouble: Double = 1.0
        var DoubleSizeBits = Double.SIZE_BITS
        var DoubleSizeByte = Double.SIZE_BYTES

        //Mostrar por pantalla
        //Dos formas de mostrar los colores del texto
        Text(text = "==================================", color=Color(0,100,100))
```

```
        Text(text = "Mostrar de variables",color=Color.Red)
        Text(text = "==================================", color=Color(0,100,100))
        Text(text = "unFloat = $unFloat")
        Text(text = "Longitud Float en Bits = $FloatSizeBits",color=Color.Blue)
        Text(text = "Longitud Float en Bytes = $FloatSizeByte",color=Color.Red)
        Text(text ="Máximo valor de Float: " + MaxFloat,color=Color.Blue)
        Text(text = "Mínimo valor de Float: $MinFloat",color=Color.Red)
        Text(text = "El tipo de la variable es: ${unFloat::class.simpleName}",color=Color.Magenta)
        Text(text = "==================================", color=Color(0,100,100))
        Text(text = "unShort = $unDouble")
        Text(text = "Longitud Float en Bits = $DoubleSizeBits",color=Color.Blue)
        Text(text = "Longitud Float en Bytes = $DoubleSizeByte",color=Color.Red)
        Text(text ="Máximo valor de Float: " + MaxDouble,color=Color.Blue)
        Text(text = "Mínimo valor de Float: $MinDouble",color=Color.Red)
        Text(text = "El tipo de la variable es: ${unDouble::class.simpleName}",color=Color.Magenta)
        Text(text = "==================================", color=Color(0,100,100))

    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_tutorial_2Theme {
        Greeting("Android")
    }
}
```

MainActivity.kt:

```
package com.example.tema2_tutorial_3

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_tutorial_3.ui.theme.Tema1_tutorial_3Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_tutorial_3Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Android")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Column {
        //lineas de código que influye la etiqueta Column
        Text(text = "Tipo $name!",color= Color.Red)

        //Declaración de variables
        //Para Char hay que poner ''
        var unChar: Char = 'a'
        var nuevaLinea: Char ='\n'//Imprimir nueva linea, carácter escapado
        var escapeUnicode: Char ='\uFF00'
        var CharSizeBits = Char.SIZE_BITS
        var CharSizeByte = Char.SIZE_BYTES
```

```
        //Mostrar por pantalla
        //Dos formas de mostrar los colores del texto
        Text(text = "==================================", color= Color(0,100,100))
        Text(text = "Mostrar de variables",color= Color.Red)
        Text(text = "==================================", color= Color(0,100,100))
        Text(text = "unChar = $unChar")
        Text(text = "Longitud Char en Bits = $CharSizeBits",color= Color.Blue)
        Text(text = "Longitud Char en Bytes = $CharSizeByte",color= Color.Red)
        Text(text ="Imprimr nueva línea"+nuevaLinea,color= Color.Blue)//Imprime un carácter de nueva
línea adicional
        Text(text = "Imprimir escape unicode: $escapeUnicode",color= Color.Red)
        Text(text = "El tipo de la variable es: ${unChar::class.simpleName}",color= Color.Magenta)
        Text(text = "==================================", color= Color(0,100,100))

    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_tutorial_3Theme {
        Greeting("Android")
    }
}
```

Cada etiqueta de bloque comienza en una nueva línea y comienza con el @carácter.

Aquí hay un ejemplo de una clase documentada usando KDoc:

```
/**
* Un grupo de  *members*.
*
* Esta clase no tiene una lógica útil; es solo un ejemplo de documentación..
*
* @param T the type of a member in this group.
* @property name the name of this group.
* @constructor Creates an empty group.
*/
class Group<T>(val name: String) {
   /**
    * Adds a [member] to this group.
    * @return the new size of the group.
    */
   fun add(member: T): Int { ... }
}
```

Además, admite el doble **//** para comentar línea a línea

```
//Declaración de variables
//Para Char hay que poner ''
var unChar: Char = 'a'
var nuevaLinea: Char ='\n'//Imprimir nueva linea, carácter escapado
var escapeUnicode: Char ='\uFF00'
var CharSizeBits = Char.SIZE_BITS
var CharSizeByte = Char.SIZE_BYTES
```

MainActivity.kt:

```
package com.example.tema2_tutorial_4

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
```

```
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_tutorial_4.ui.theme.Tema1_tutorial_4Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_tutorial_4Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Android")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Column {
        //líneas de código que influye la etiqueta Column
        Text(text = "Tipo $name!",color= Color.Red)

        //Declaración de variables
        var resultadoSuma: Int = 0
        var resultadoResta: Int = 0
        var resultadoMul: Int = 0
        var resultadoDiv: Int = 0
        var resultadoModulo: Int = 0
        var num1: Int = 15
        var num2: Int = 3

        //Operaciones
        resultadoSuma = num1 + num2
        resultadoResta = num1 - num2
        resultadoMul = num1 * num2
        resultadoDiv = num1 / num2
        resultadoModulo = num1 % num2

        //Mostrar por pantalla
        //Dos formas de mostrar los colores del texto
        Text(text = "==================================", color= Color(0,100,100))
        Text(text = "Mostrar resultados",color= Color.Red)
        Text(text = "==================================", color= Color(0,100,100))
        Text(text ="Resultado Suma: "+resultadoSuma+"="+num1+"+"+num2,color= Color.Blue)
        Text(text = "Resultado Resta: $resultadoResta = $num1 - $num2",color= Color.Red)
        Text(text ="Resultado Multiplicación: "+resultadoMul+"="+num1+"*"+num2,color= Color.Magenta)
        Text(text = "Resultado División: $resultadoDiv = $num1 / $num2",color= Color.Cyan)
        Text(text = "Resultado Módulo(resto): $resultadoModulo = $num1 / $num2",color= Color.Yellow)
        Text(text = "==================================", color= Color(0,100,100))

    }
}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_tutorial_4Theme {
        Greeting("Android")
    }
}
```

2.6 Problemas con solucionario

2.6.1. Realiza un programa que muestre tus deportes favoritos por orden

Ordinograma

MainActivity.kt:

```
package com.example.tema2_problema_1

import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_problema_1.ui.theme.Tema1_problema_1Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_1Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Deportes favoritos")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Column {
        Text(text = "$name!")
        Text(text = "1. Patinaje")
        Text(text = "2. Vela")
        Text(text = "3. Golf")
    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_1Theme {
        Greeting("Android")
    }
}
```

2.6.2. Crea un programa que a partir de la medida en pulgadas la transforme en su equivalente de centímetros

MainActivity.kt:

```
package com.example.tema2_problema_2

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_problema_2.ui.theme.Tema1_problema_2Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_2Theme {
                // A surface container using the 'background' color from the theme
```

```
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Conversión de pulgada a cm")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Column {
        Text(text = "$name!")

        //Declaración constante
        //Valor constante de la pulgada anglosajona
        val const_pulgada: Double = 2.54

        //Declaración de variables
        var medida_pulgada: Double = 25.0

        var resultado_centimetros: Double = 0.0

        //Operaciones
        resultado_centimetros = medida_pulgada * const_pulgada

        //Mostrar por pantalla
        Text(text ="De Pulgadas: "+medida_pulgada+"a Centímetros:"+resultado_centimetros,color=
Color.Blue)
    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_2Theme {
        Greeting("Android")
    }
}
```

Partes nuevas introducidas:

```
OutlinedTextField (
   value = medida_pulgada,
   onValueChange = { medida_pulgada = it },
   label = {
       Text("Medida en pulgadas")
   },
   modifier = Modifier
       .fillMaxWidth()
       .padding(10.dp),
   singleLine = true
)
```

El estado en elementos que admiten composición

```
//Declaración de variables
var medida_pulgada by remember { mutableStateOf("") }
var  resultado_centimetros by remember { mutableStateOf("") }
```

MainActivity.kt:

```
package com.example.tema2_problema_3

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
```

```
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.tema2_problema_3.ui.theme.Tema1_problema_3Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_3Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Conversión de pulgada a cm")
                }
            }
        }
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column(modifier = Modifier.padding(20.dp)) {
        Text(text = "$name",textAlign = TextAlign.Center,color= Color.Green,
            fontWeight = FontWeight.Bold)
    }

    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center
    )
    {

        //Declaración constante
        //Valor constante de la pulgada anglosajona
        val const_pulgada: Double = 2.54

        //Declaración de variables
        var medida_pulgada by remember { mutableStateOf("") }
        var  resultado_centimetros by remember { mutableStateOf("") }

        OutlinedTextField (
            value = medida_pulgada,
            onValueChange = { medida_pulgada = it },
            label = {
                Text("Medida en pulgadas")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )

        Button(
            onClick = {
                //Operaciones
                resultado_centimetros = (medida_pulgada.toDouble() * const_pulgada).toString()
            },
            modifier = Modifier.padding(10.dp)
        ) {
```

```
            Text(text = "Calcular")
        }

        //Mostrar por pantalla
        Text(
            text ="De Pulgadas: "+medida_pulgada+" a Centímetros: "+resultado_centimetros,
            color= Color.Blue,
            modifier = Modifier.padding(10.dp)
        )
    }


}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_3Theme {
        Greeting("Android")
    }
}
```

MainActivity.kt:

```
package com.example.tema2_problema_4

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_problema_4.ui.theme.Tema2_problema_4Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_4Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Cálculo de la longitud y área del círculo")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Column {
        Text(text = "Hello $name!")
        Text(text = "====================================",color= Color.Green)

        //Declaración de variables
        var radio: Double = 30.0
        var longitud: Double = 0.0
        var area: Double = 0.0


        //Operaciones
        //utilizar la constante PI  que ya proporciona Kotlin
        longitud =  kotlin.math.PI * 2* radio

        area = kotlin.math.PI * radio * radio
```

```
        //Mostrar por pantalla
        Text(text ="Longitud circunferencia: "+longitud,color= Color.Blue)
        Text(text ="Área círculo: "+area,color= Color.Red)
    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_4Theme {
        Greeting("Android")
    }
}
```

MainActivity.kt:

```
package com.example.tema2_problema_5

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.tema2_problema_5.ui.theme.Tema2_problema_5Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_5Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Cálculo de la longitud y área del círculo")
                }
            }
        }
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column(modifier = Modifier.padding(20.dp)) {
        Text(text = "$name",textAlign = TextAlign.Center,color= Color.Green,
            fontWeight = FontWeight.Bold)
    }

    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center
    )
    {

        //Declaración de variables
```

```
        var radio by remember { mutableStateOf("") }
        var longitud by remember { mutableStateOf("") }
        var area  by remember { mutableStateOf("") }

        OutlinedTextField (
            value = radio,
            onValueChange = { radio = it },
            label = {
                Text("Radio en centímetros")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )

        Button(
            onClick = {
                //Operaciones
                longitud = (kotlin.math.PI * 2 * radio.toDouble() ).toString()
                area = (kotlin.math.PI * radio.toDouble() * radio.toDouble()).toString()
            },
            modifier = Modifier.padding(10.dp)
        ) {
            Text(text = "Calcular")
        }

        //Mostrar por pantalla
        Text(
            text ="Longitud circunferencia: "+longitud,
            color= Color.Blue,
            modifier = Modifier.padding(10.dp)
        )

        Text(
            text ="Área círculo: "+area,
            color= Color.Red,
            modifier = Modifier.padding(10.dp)
        )
    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_5Theme {
        Greeting("Android")
    }
}
```

2.6.4. Diseñar un programa que pida la altura del triángulo y calcule su área

MainActivity.kt:

```
package com.example.tema2_problema_6

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
```

```
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_problema_6.ui.theme.Tema2_problema_6Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_6Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Cálculo del área del triángulo")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column {
        Text(text = "Hello $name!")
        Text(text = "====================================",color= Color.Green)

        //Declaración de variables
        var altura: Double = 15.0
        var base: Double = 10.0
        var area: Double = 0.0


        //Operaciones
        //calcular el área del triángulo

        area = base * altura / 2.0

        //Mostrar por pantalla
        Text(text ="Área triángulo: "+area,color= Color.Red)
    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_6Theme {
        Greeting("Android")
    }
}
```

MainActivity.kt:

```
package com.example.tema2_problema_7

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
```

```
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.tema2_problema_7.ui.theme.Tema2_problema_7Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_7Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Cálculo del área del triángulo")
                }
            }
        }
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column(modifier = Modifier.padding(20.dp)) {
        Text(text = "$name",textAlign = TextAlign.Center,color= Color.Green,
            fontWeight = FontWeight.Bold)
    }

    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center
    )
    {

        //Declaración de variables
        var altura by remember { mutableStateOf("") }
        var base by remember { mutableStateOf("") }
        var area  by remember { mutableStateOf("") }

        OutlinedTextField (
            value = altura,
            onValueChange = { altura = it },
            label = {
                Text("Altura en centímetros")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )

        OutlinedTextField (
            value = base,
            onValueChange = { base = it },
            label = {
                Text("Base en centímetros")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )

        Button(
            onClick = {
                //Operaciones
                area = (altura.toDouble() * base.toDouble() / 2.0).toString()
            },
```

```
            modifier = Modifier.padding(10.dp)
        ) {
            Text(text = "Calcular")
        }

        //Mostrar por pantalla
        Text(
            text ="Área del triángulo: "+area,
            color= Color.Blue,
            modifier = Modifier.padding(10.dp)
        )

    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_7Theme {
        Greeting("Android")
    }
}
```

2.6.5. Realiza un programa que pida la temperatura en la escala Celsius, los devuelva en Fahrenheit y Kelvin

MainActivity.kt:

```
package com.example.tema2_problema_8

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_problema_8.ui.theme.Tema2_problema_8Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_8Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Conversión Celsius a Fahrenheit y Kelvin")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column {
        Text(text = "$name!")
        Text(text = "======================================",color= Color.Green)

        //Declaración de variables
        var celsius: Double = 15.0
        var farhrenheit: Double = 0.0
        var kelvin: Double = 0.0


        //Operaciones
        //calcular el cambio de escala Fahrenheit y Kelvin
```

```
        farhrenheit = (celsius * 9.0 / 5.0) + 32 //Fahrenheit
        kelvin = celsius + 273.15 //Kelvin


        //Mostrar por pantalla
        Text(text ="Temperatura Celsius: "+celsius,color= Color.Red)
        Text(text ="Temperatura Fahrenheit: "+farhrenheit,color= Color.Blue)
        Text(text ="Temperatura Kelvin: "+kelvin,color= Color.Magenta)
    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_8Theme {
        Greeting("Android")
    }
}
```

Para darle más versatilidad al programa se podrá introducir por pantalla la temperatura en grados Celsius.

MainActivity.kt:

```
package com.example.tema2_problema_9

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.tema2_problema_9.ui.theme.Tema2_problema_9Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_9Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Conversión Celsius a Fahrenheit y Kelvin")
                }
            }
        }
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
```

```
    Column(modifier = Modifier.padding(20.dp)) {
        Text(text = "$name",textAlign = TextAlign.Center,color= Color.Green,
            fontWeight = FontWeight.Bold)
    }

    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center
    )
    {

        //Declaración de variables
        var celsius by remember { mutableStateOf("") }
        var farhrenheit by remember { mutableStateOf("") }
        var kelvin  by remember { mutableStateOf("") }

        OutlinedTextField (
            value = celsius,
            onValueChange = { celsius = it },
            label = {
                Text("Tempartura en grados Celsius")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )

        Button(
            onClick = {
                //Operaciones
                //calcular el cambio de escala Fahrenheit y Kelvin

                farhrenheit = ((celsius.toDouble() * 9.0 / 5.0) + 32).toString() //Fahrenheit
                kelvin = (celsius.toDouble() + 273.15).toString() //Kelvin
            },
            modifier = Modifier.padding(10.dp)
        ) {
            Text(text = "Calcular")
        }

        //Mostrar por pantalla
        Text(
            text ="Temperatura Fahrenheit: "+farhrenheit,
            color= Color.Blue,
            modifier = Modifier.padding(10.dp)
        )
        Text(
            text ="Temperatura Kelvin: "+kelvin,
            color= Color.Magenta,
            modifier = Modifier.padding(10.dp)
        )

    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_9Theme {
        Greeting("Android")
    }
}
```

2.6.6. Uso de las funciones del cálculo matemático en kotlin

MainActivity.kt:

```
package com.example.tema2_problema_10

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
```

```
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_problema_10.ui.theme.Tema2_problema_10Theme
import kotlin.math.E
import kotlin.math.PI
import kotlin.math.absoluteValue
import kotlin.math.ceil
import kotlin.math.cos
import kotlin.math.floor
import kotlin.math.pow
import kotlin.math.sign
import kotlin.math.sqrt

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_10Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Funciones y constantes matemáticas")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column {
        Text(text = "$name!")
        Text(text = "====================================",color= Color.Green)

        //Declaración de variables
        var doble: Double = -15.5
        var flotante: Float = -50.6F
        var entero: Int = -20
        var largo: Long = -30
        var dato: Double = 81.0

        //Mostrar por pantalla
        Text(text ="Valor original del Double: "+doble,color= Color.Red)
        Text(text ="Valor absoluto del Double: "+doble.absoluteValue,color= Color.Blue)
        Text(text ="Valor original del Float: "+flotante,color= Color.Red)
        Text(text ="Valor absoluto del Float: "+flotante.absoluteValue,color= Color.Blue)
        Text(text ="Valor original del Integer: "+entero,color= Color.Red)
        Text(text ="Valor absoluto del Integer: "+entero.absoluteValue,color= Color.Blue)
        Text(text ="Valor original del Long: "+largo,color= Color.Red)
        Text(text ="Valor absoluto del Long: "+largo.absoluteValue,color= Color.Blue)
        Text(text = "====================================",color= Color.Green)
        Text(text ="Valor de Pi: "+ PI,color= Color.Red)
        Text(text ="Valor de E: "+ E,color= Color.Blue)
        Text(text = "====================================",color= Color.Green)
        Text(text ="Valor de Double: "+doble,color= Color.Red)
        Text(text ="Signo de Double para esa variable: "+doble.sign,color= Color.Blue)
        Text(text ="Coseno: "+cos(doble),color= Color.Red)
        Text(text ="Redondeo: "+ceil(doble),color= Color.Blue)
        Text(text ="Redondea el valor de doble a número entero hacia el infinito negativo:
"+floor(doble),color= Color.Red)
        Text(text = "====================================",color= Color.Green)
        Text(text ="Potencia de ${dato}: "+dato.pow(2),color= Color.Blue)
        Text(text ="Raiz Cuadrada de ${dato}: "+sqrt(dato),color= Color.Red)

    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_10Theme {
        Greeting("Android")
    }
}
```

2.6.7. Escribir un programa que calcule el área de un rectángulo

MainActivity.kt:

```
package com.example.tema2_problema_11

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_problema_11.ui.theme.Tema2_problema_11Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_11Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Cálculo del área del rectángulo")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column {
        Text(text = "$name!")
        Text(text = "====================================",color= Color.Green)

        //Declaración de variables
        var largo: Double = 15.0
        var ancho: Double = 10.0
        var area: Double = 0.0

        //Operaciones
        //calcular el aŕea del rectángulo

        area = largo * ancho

        //Mostrar por pantalla
        Text(text ="Área rectángulo: "+area,color= Color.Red)
    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_11Theme {
        Greeting("Android")
    }
}
```

Diseñar una interfaz para que pueda recibir los datos que le pase el usuario:

MainActivity.kt:

```
package com.example.tema2_problema_12

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.tema2_problema_12.ui.theme.Tema2_problema_12Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_12Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Cálculo del área del rectángulo")
                }
            }
        }
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column(modifier = Modifier.padding(20.dp)) {
        Text(text = "$name",textAlign = TextAlign.Center,color= Color.Green,
            fontWeight = FontWeight.Bold)
    }

    Column(
        modifier = Modifier.fillMaxSize(),
```

```
        verticalArrangement = Arrangement.Center
    )
    {


        //Declaración de variables
        var largo by remember { mutableStateOf("") }
        var ancho by remember { mutableStateOf("") }
        var area  by remember { mutableStateOf("") }

        OutlinedTextField (
            value = largo,
            onValueChange = { largo = it },
            label = {
                Text("Largo en centímetros")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )

        OutlinedTextField (
            value = ancho,
            onValueChange = { ancho = it },
            label = {
                Text("Ancho en centímetros")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )

        Button(
            onClick = {
                //Operaciones
                area = (largo.toDouble() * ancho.toDouble()).toString()
            },
            modifier = Modifier.padding(10.dp)
        ) {
            Text(text = "Calcular")
        }

        //Mostrar por pantalla
        Text(
            text ="Área del rectángulo: "+area,
            color= Color.Blue,
            modifier = Modifier.padding(10.dp)
        )

    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_12Theme {
        Greeting("Android")
    }
}
```

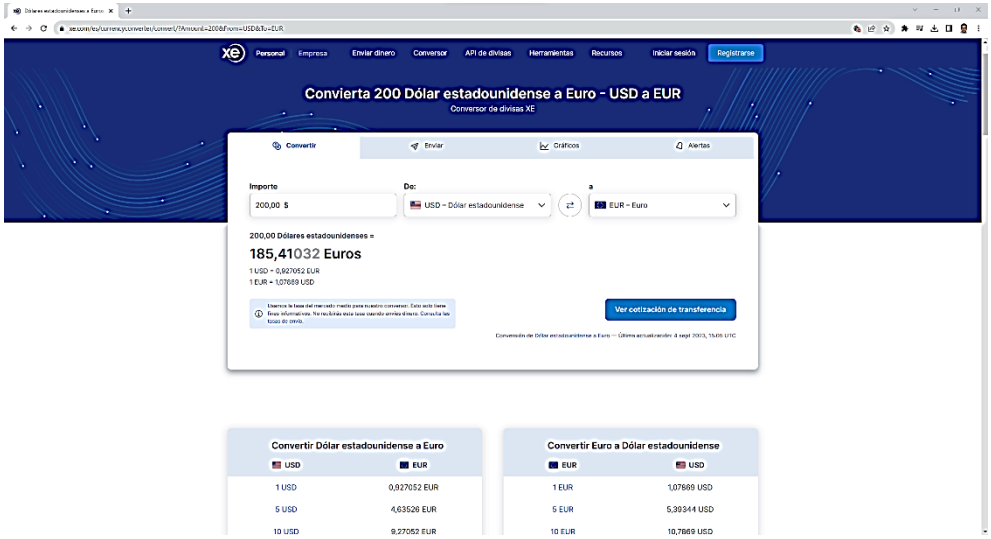
2.6.8. Realizar un programa que convierta de dólares a euros

MainActivity.kt:

```
package com.example.tema2_problema_13
```

```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_problema_13.ui.theme.Tema2_problema_13Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_13Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Conversión de Dólares a Euros")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column {
        Text(text = "$name!")
        Text(text = "====================================",color= Color.Green)

        //Declaración de variables
        var cotizacion: Double = 0.909570
        var dolares: Double = 200.0
        var euros: Double = 0.0


        //Operaciones
        //convertir dólares

        euros = dolares * cotizacion

        //Mostrar por pantalla
        Text(text ="Cantidad de $dolares\$ equivale a " +euros+"€",color= Color.Red)
    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_13Theme {
        Greeting("Android")
    }
}
```

Mirar la cotización en tiempo real en:
https://www.xe.com/currencyconverter/



MainActivity.kt:

```
package com.example.tema2_problema_14

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.tema2_problema_14.ui.theme.Tema2_problema_14Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_14Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Conversión de Dólares a Euros")
                }
            }
        }
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
```

```
    Column(modifier = Modifier.padding(20.dp)) {
        Text(text = "$name",textAlign = TextAlign.Center,color= Color.Green,
            fontWeight = FontWeight.Bold)
    }
    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center
    )
    {

        //Declaración de variables
        var cotizacion by remember { mutableStateOf("") }
        var dolares by remember { mutableStateOf("") }
        var euros  by remember { mutableStateOf("") }

        OutlinedTextField (
            value = cotizacion,
            onValueChange = { cotizacion = it },
            label = {
                Text("Cotización actual Dólar")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )

        OutlinedTextField (
            value = dolares,
            onValueChange = { dolares = it },
            label = {
                Text("Cantidad Dólares")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )

        Button(
            onClick = {
                //Operaciones
                euros = (cotizacion.toDouble() * dolares.toDouble()).toString()
            },
            modifier = Modifier.padding(10.dp)
        ) {
            Text(text = "Calcular")
        }

        //Mostrar por pantalla

        Text(
            text ="Cantidad de $dolares\$ equivale a " +euros+"€",
            color= Color.Blue,
            modifier = Modifier.padding(10.dp)
        )

    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_14Theme {
        Greeting("Android")
    }
}
```

2.6.9. Crear un programa que calcule la nota media de 3 exámenes

MainActivity.kt:

```
package com.example.tema2_problema_15

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_problema_15.ui.theme.Tema2_problema_15Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_15Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Media de 3 notas")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column {
        Text(text = "$name!")
        Text(text = "====================================",color= Color.Green)

        //Declaración de variables
        var nota1: Double = 5.0
        var nota2: Double = 8.0
        var nota3: Double = -1.0
        var media: Double = 0.0


        //Operaciones
        //calcular la media de tres notas

        media = (nota1 + nota2 + nota3)/3.0

        //Mostrar por pantalla
        Text(text ="Nota Media de $nota1, $nota2, $nota3 es: "+media,color= Color.Red)
    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_15Theme {
        Greeting("Android")
    }
}
```

Segunda parte: creación de un sistema gráfico para poder el usuario introducir las notas que desee.

MainActivity.kt:

```
package com.example.tema2_problema_16

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.tema2_problema_16.ui.theme.Tema2_problema_16Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_16Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Media de 3 notas")
                }
            }
        }
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column(modifier = Modifier.padding(20.dp)) {
        Text(text = "$name",textAlign = TextAlign.Center,color= Color.Green,
            fontWeight = FontWeight.Bold)
    }

    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center
    )
    {

        //Declaración de variables
        var nota1 by remember { mutableStateOf("") }
        var nota2 by remember { mutableStateOf("") }
        var nota3  by remember { mutableStateOf("") }
        var media  by remember { mutableStateOf("") }

        OutlinedTextField (
            value = nota1,
            onValueChange = { nota1 = it },
```

```
            label = {
                Text("Introducir Nota1")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )

        OutlinedTextField (
            value = nota2,
            onValueChange = { nota2 = it },
            label = {
                Text("Introducir Nota2")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )

        OutlinedTextField (
            value = nota3,
            onValueChange = { nota3 = it },
            label = {
                Text("Introducir Nota3")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )

        Button(
            onClick = {
                //Operaciones
                media = ((nota1.toDouble() + nota2.toDouble() + nota3.toDouble())/3.0).toString()
            },
            modifier = Modifier.padding(10.dp)
        ) {
            Text(text = "Calcular")
        }

        //Mostrar por pantalla

        Text(
            text = "Nota Media de $nota1, $nota2, $nota3 es: "+media,
            color= Color.Blue,
            modifier = Modifier.padding(10.dp)
        )

    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_16Theme {
        Greeting("Android")
    }
}
```

2.6.10. Realizar un programa que pida al usuario el coste por sms y la cantidad de mensajes calcule el coste de envío

MainActivity.kt:

```
package com.example.tema2_problema_17

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
```

```
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_problema_17.ui.theme.Tema2_problema_17Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_17Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Cálculo coste SMS")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column {
        Text(text = "$name!")
        Text(text = "====================================",color= Color.Green)

        //Declaración de variables
        var coste_sms: Double = 0.5
        var numero_sms: Double = 10.0
        var gasto: Double = 0.0


        //Operaciones
        //calcular el gasto de los sms

        gasto = coste_sms * numero_sms

        //Mostrar por pantalla
        Text(text ="Gasto de SMS es: "+gasto,color= Color.Red)
    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_17Theme {
        Greeting("Android")
    }
}
```

Segunda parte: creación de un sistema gráfico para poder el usuario introducir el coste sms/unidad y los sms que desea enviar.

MainActivity.kt:

```
package com.example.tema2_problema_18

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
```

```
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.tema2_problema_18.ui.theme.Tema2_problema_18Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_18Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Cálculo coste SMS")
                }
            }
        }
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column(modifier = Modifier.padding(20.dp)) {
        Text(text = "$name",textAlign = TextAlign.Center,color= Color.Green,
            fontWeight = FontWeight.Bold)
    }
    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center
    )
    {

        //Declaración de variables
        var coste_sms by remember { mutableStateOf("") }
        var numero_sms by remember { mutableStateOf("") }
        var gasto  by remember { mutableStateOf("") }

        OutlinedTextField (
            value = coste_sms,
            onValueChange = { coste_sms = it },
            label = {
                Text("Introducir coste SMS unidad")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )

        OutlinedTextField (
            value = numero_sms,
            onValueChange = { numero_sms = it },
            label = {
                Text("Número SMS enviar")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
```

```
        )

        Button(
            onClick = {
                //Operaciones
                gasto = (coste_sms.toDouble() * numero_sms.toDouble()).toString()
            },
            modifier = Modifier.padding(10.dp)
        ) {
            Text(text = "Calcular")
        }

        //Mostrar por pantalla

        Text(
            text = "Gasto de SMS es: "+gasto,
            color= Color.Blue,
            modifier = Modifier.padding(10.dp)
        )

    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_18Theme {
        Greeting("Android")
    }
}
```

2.6.11. Escribir un programa que pida al usuario una cantidad en segundos y devuelva las horas, minutos y segundos que contendrá

Primera parte: comprensión del problema que se indica.

MainActivity.kt:

```
package com.example.tema2_problema_19

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_problema_19.ui.theme.Tema2_problema_19Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_19Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Convertir un cantidad e segundos en Horas, Minutos y Segundos")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column {
```

```
        Text(text = "$name!")
        Text(text = "====================================",color= Color.Green)

        //Declaración constantes
        val sexagesimal: Int = 60

        //Declaración de variables
        var cantidad_segundos: Int = 365137
        var horas: Int = 0
        var minutos: Int = 0
        var resto_seg: Int = 0
        var resto_min: Int = 0


        //Operaciones

        //calcular los segundos que quedarán
        minutos = cantidad_segundos / sexagesimal
        resto_seg = cantidad_segundos % sexagesimal

        //calcular los minutos que quedarán
        horas = minutos / sexagesimal
        resto_min = minutos % sexagesimal



        //Mostrar por pantalla
        Text(text ="cantidad segundos a convertir: "+cantidad_segundos,color= Color.Red)
        Text(text ="Horas: "+horas,color= Color.Blue)
        Text(text ="Minutos: "+resto_min,color= Color.Red)
        Text(text ="Segundos: "+resto_seg,color= Color.Blue)
    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_19Theme {
        Greeting("Android")
    }
}
```

Segunda parte: creación de un sistema gráfico para poder el usuario introducir la cantidad de segundos que desee el usuario.

MainActivity.kt:

```
package com.example.tema2_problema_20

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.tema2_problema_20.ui.theme.Tema2_problema_20Theme
```

```
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_20Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Convertir un cantidad e segundos en Horas, Minutos y Segundos")
                }
            }
        }
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column(modifier = Modifier.padding(20.dp)) {
        Text(text = "$name",textAlign = TextAlign.Center,color= Color.Green,
            fontWeight = FontWeight.Bold)
    }

    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center
    )
    {
        //Declaración constantes
        val sexagesimal: Int = 60

        //Declaración de variables
        var cantidad_segundos by remember { mutableStateOf("") }
        var horas by remember { mutableStateOf("") }
        var minutos by remember { mutableStateOf("") }
        var resto_seg by remember { mutableStateOf("") }
        var resto_min by remember { mutableStateOf("") }


        OutlinedTextField (
            value = cantidad_segundos,
            onValueChange = { cantidad_segundos = it },
            label = {
                Text("Introducir cantidad segundos a convertir")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )


        Button(
            onClick = {
                //Operaciones
                //calcular los segundos que quedarán
                minutos = (cantidad_segundos.toInt() / sexagesimal).toString()
                resto_seg = (cantidad_segundos.toInt() % sexagesimal).toString()

                //calcular los minutos que quedarán
                horas = (minutos.toInt() / sexagesimal).toString()
                resto_min = (minutos.toInt() % sexagesimal).toString()

            },
            modifier = Modifier.padding(10.dp)
        ) {
            Text(text = "Calcular")
        }

        //Mostrar por pantalla

        Text(
            text = "Cantidad segundos a convertir: "+cantidad_segundos,
            color= Color.Red,
            modifier = Modifier.padding(10.dp)
        )
        Text(
            text = "Horas: "+horas,
```

```
            color= Color.Blue,
            modifier = Modifier.padding(10.dp)
        )
        Text(
            text = "Minutos: "+resto_min,
            color= Color.Red,
            modifier = Modifier.padding(10.dp)
        )
        Text(
            text = "Segundos: "+resto_seg,
            color= Color.Blue,
            modifier = Modifier.padding(10.dp)
        )

    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_20Theme {
        Greeting("Android")
    }
}
```

2.6.12. Realizar un programa que calcule el volumen de la esfera

MainActivity.kt:

```
package com.example.tema2_problema_21

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_problema_21.ui.theme.Tema2_problema_21Theme
import kotlin.math.PI
import kotlin.math.pow

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_21Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Cálculo volumen esfera")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column {
        Text(text = "$name!")
        Text(text = "====================================",color= Color.Green)


        //Declaración de variables
        var volumen: Double = 0.0
```

```
        var radio: Double = 5.0


        //Operaciones

        //calcular los segundos que quedarán
        volumen = 4* PI * radio.pow(3.0) / 3.0

        //Mostrar por pantalla
        Text(text ="Volumen de la esfera de radio $radio: "+volumen,color= Color.Red)
    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_21Theme {
        Greeting("Android")
    }
}
```

Segunda parte: creación de un sistema gráfico para poder el usuario introducir el radio de la esfera para reutilizar el código cuantas veces sea necesario.

MainActivity.kt:

```
package com.example.tema2_problema_22

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.tema2_problema_22.ui.theme.Tema2_problema_22Theme
import kotlin.math.PI
import kotlin.math.pow

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_22Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Cálculo volumen esfera")
                }
            }
        }
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
```

```
    Column(modifier = Modifier.padding(20.dp)) {
        Text(text = "$name",textAlign = TextAlign.Center,color= Color.Green,
            fontWeight = FontWeight.Bold)
    }
    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center
    )
    {

        //Declaración de variables
        var volumen by remember { mutableStateOf("") }
        var radio by remember { mutableStateOf("") }



        OutlinedTextField (
            value = radio,
            onValueChange = { radio = it },
            label = {
                Text("Introducir radio esfera en centímetros")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )


        Button(
            onClick = {
                //Operaciones
                //calcular volumen esfera
                volumen = (4.0 * PI * radio.toDouble().pow(3.0) / 3.0).toString()

            },
            modifier = Modifier.padding(10.dp)
        ) {
            Text(text = "Calcular")
        }


        //Mostrar por pantalla

        Text(
            text = "Volumen de la esfera de radio $radio: "+volumen,
            color= Color.Red,
            modifier = Modifier.padding(10.dp)
        )


    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_22Theme {
        Greeting("Android")
    }
}
```

2.6.13. Codificar un programa que calcule el área y volumen del cilindro

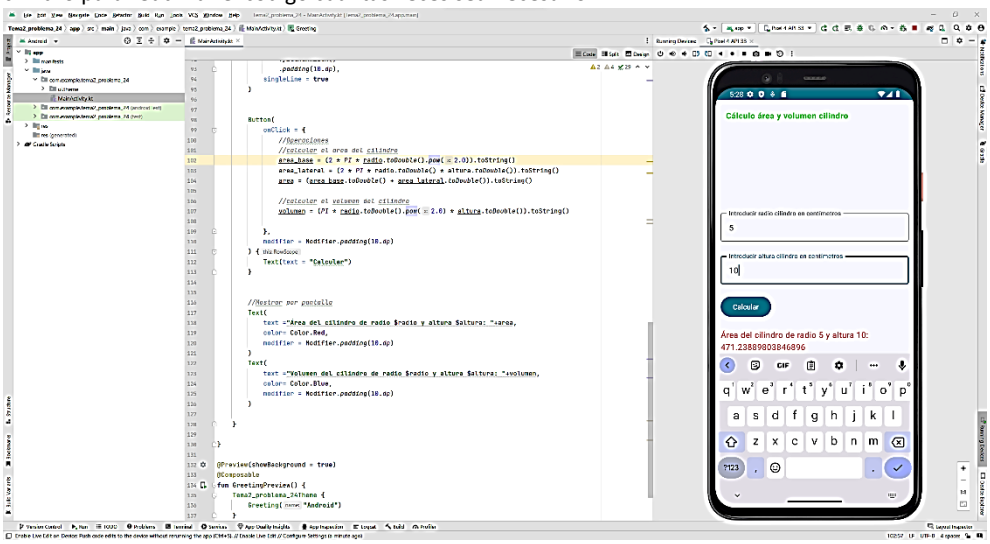
Primera parte: comprensión del problema que se indica.

MainActivity.kt:

```
package com.example.tema2_problema_23

import android.os.Bundle
import androidx.activity.ComponentActivity
```

```
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_problema_23.ui.theme.Tema2_problema_23Theme
import kotlin.math.PI
import kotlin.math.pow

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_23Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Cálculo área y volumen cilindro")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column {
        Text(text = "$name!")
        Text(text = "====================================",color= Color.Green)


        //Declaración de variables
        var area_base: Double = 0.0
        var area_lateral: Double = 0.0
        var area: Double = 0.0
        var volumen: Double = 0.0
        var radio: Double = 5.0
        var altura: Double = 10.0


        //Operaciones

        //calcular el area del cilindro
        area_base = 2 * PI * radio.pow(2.0)
        area_lateral = 2 * PI * radio * altura
        area = area_base + area_lateral

        //calcular el volumen del cilindro
        volumen = PI * radio.pow(2.0) * altura

        //Mostrar por pantalla
        Text(text ="Área del cilindro de radio $radio y altura $altura: "+area,color= Color.Red)
        Text(text ="Volumen del cilindro de radio $radio y altura $altura: "+volumen,color=
Color.Blue)
    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_23Theme {
        Greeting("Android")
    }
}
```

Segunda parte: creación de un sistema gráfico para poder el usuario introducir el radio y la altura del cilindro para reutilizar el código cuantas veces sea necesario.



MainActivity.kt:

```
package com.example.tema2_problema_24

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.tema2_problema_24.ui.theme.Tema2_problema_24Theme
import kotlin.math.PI
import kotlin.math.pow

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_24Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Cálculo área y volumen cilindro")
                }
            }
        }
```

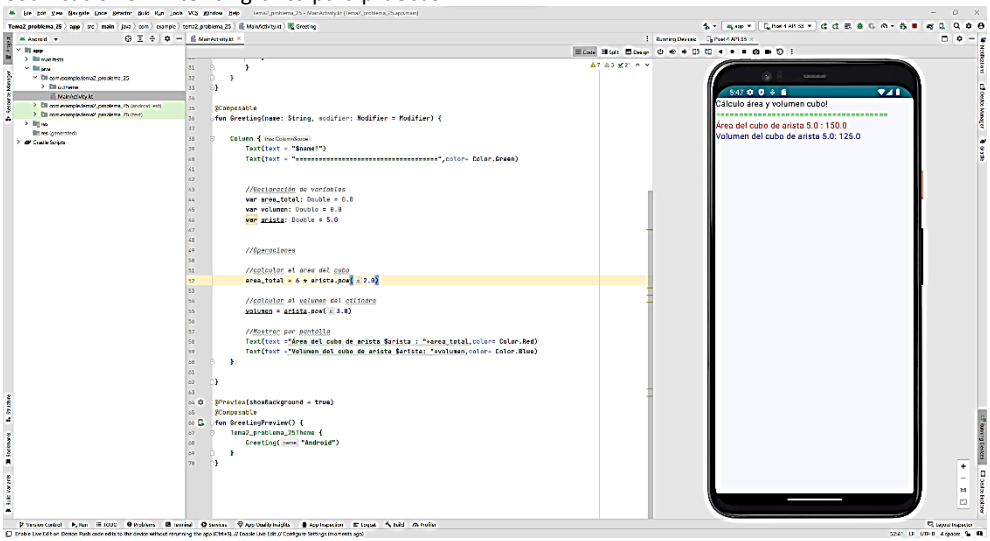
```
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column(modifier = Modifier.padding(20.dp)) {
        Text(text = "$name",textAlign = TextAlign.Center,color= Color.Green,
            fontWeight = FontWeight.Bold)
    }

    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center
    )
    {

        //Declaración de variables

        var area_base by remember { mutableStateOf("") }
        var area_lateral by remember { mutableStateOf("") }
        var area by remember { mutableStateOf("") }
        var volumen by remember { mutableStateOf("") }
        var radio by remember { mutableStateOf("") }
        var altura by remember { mutableStateOf("") }

        OutlinedTextField (
            value = radio,
            onValueChange = { radio = it },
            label = {
                Text("Introducir radio cilindro en centímetros")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )

        OutlinedTextField (
            value = altura,
            onValueChange = { altura = it },
            label = {
                Text("Introducir altura cilindro en centímetros")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )


        Button(
            onClick = {
                //Operaciones
                //calcular el area del cilindro
                area_base = (2 * PI * radio.toDouble().pow(2.0)).toString()
                area_lateral = (2 * PI * radio.toDouble() * altura.toDouble()).toString()
                area = (area_base.toDouble() + area_lateral.toDouble()).toString()

                //calcular el volumen del cilindro
                volumen = (PI * radio.toDouble().pow(2.0) * altura.toDouble()).toString()

            },
            modifier = Modifier.padding(10.dp)
        ) {
            Text(text = "Calcular")
        }


        //Mostrar por pantalla
        Text(
            text ="Área del cilindro de radio $radio y altura $altura: "+area,
            color= Color.Red,
            modifier = Modifier.padding(10.dp)
        )
        Text(
            text ="Volumen del cilindro de radio $radio y altura $altura: "+volumen,
            color= Color.Blue,
            modifier = Modifier.padding(10.dp)
        )
```

```
    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_24Theme {
        Greeting("Android")
    }
}
```

2.6.14. Realizar un programa que calcule el área y volumen del cubo

Primera parte: comprensión del problema que se indica.

Codificación sin interfaz gráfica para pruebas.

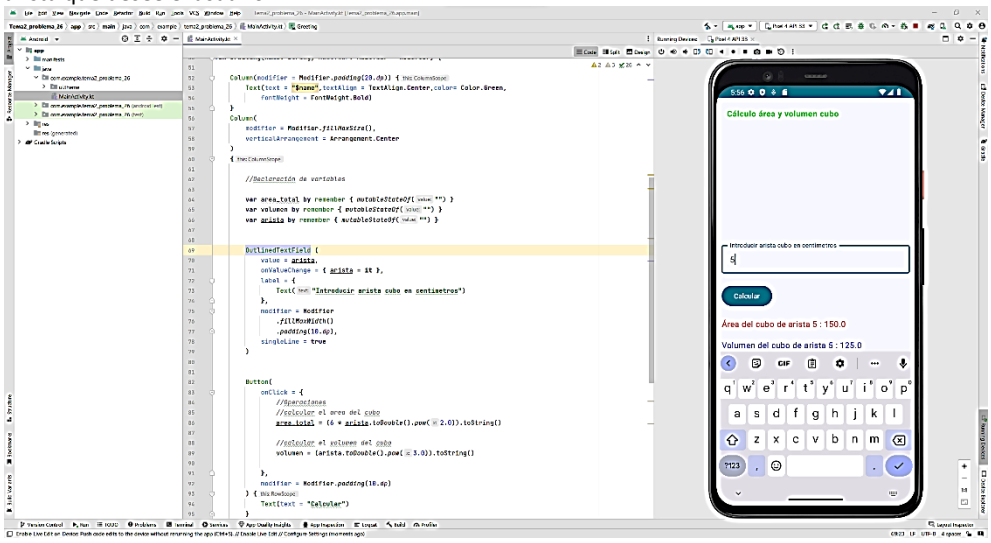


MainActivity.kt:

```
package com.example.tema2_problema_25

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import com.example.tema2_problema_25.ui.theme.Tema2_problema_25Theme
import kotlin.math.pow

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_25Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Cálculo área y volumen cubo")
```

```
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column {
        Text(text = "$name!")
        Text(text = "====================================",color= Color.Green)


        //Declaración de variables
        var area_total: Double = 0.0
        var volumen: Double = 0.0
        var arista: Double = 5.0


        //Operaciones

        //calcular el area del cubo
        area_total = 6 * arista.pow(2.0)

        //calcular el volumen del cilindro
        volumen = arista.pow(3.0)

        //Mostrar por pantalla
        Text(text ="Área del cubo de arista $arista : "+area_total,color= Color.Red)
        Text(text ="Volumen del cubo de arista $arista: "+volumen,color= Color.Blue)
    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_25Theme {
        Greeting("Android")
    }
}
```

Segunda parte: creación de un sistema gráfico para poder el usuario introducir la longitud de la arista que desee el usuario.



MainActivity.kt:

```
package com.example.tema2_problema_26

import android.os.Bundle
```

```
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.tema2_problema_26.ui.theme.Tema2_problema_26Theme
import kotlin.math.pow

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_26Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting("Cálculo área y volumen cubo")
                }
            }
        }
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column(modifier = Modifier.padding(20.dp)) {
        Text(text = "$name",textAlign = TextAlign.Center,color= Color.Green,
            fontWeight = FontWeight.Bold)
    }
    Column(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center
    )
    {

        //Declaración de variables

        var area_total by remember { mutableStateOf("") }
        var volumen by remember { mutableStateOf("") }
        var arista by remember { mutableStateOf("") }


        OutlinedTextField (
            value = arista,
            onValueChange = { arista = it },
            label = {
                Text("Introducir arista cubo en centímetros")
            },
            modifier = Modifier
                .fillMaxWidth()
                .padding(10.dp),
            singleLine = true
        )


        Button(
            onClick = {
                //Operaciones
                //calcular el area del cubo
```

```
                area_total = (6 * arista.toDouble().pow(2.0)).toString()

                //calcular el volumen del cubo
                volumen = (arista.toDouble().pow(3.0)).toString()

            },
            modifier = Modifier.padding(10.dp)
        ) {
            Text(text = "Calcular")
        }


        //Mostrar por pantalla
        Text(
            text ="Área del cubo de arista $arista : "+area_total,
            color= Color.Red,
            modifier = Modifier.padding(10.dp)
        )
        Text(
            text ="Volumen del cubo de arista $arista : "+volumen,
            color= Color.Blue,
            modifier = Modifier.padding(10.dp)
        )

    }

}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    Tema2_problema_26Theme {
        Greeting("Android")
    }
}
```

ANEXO

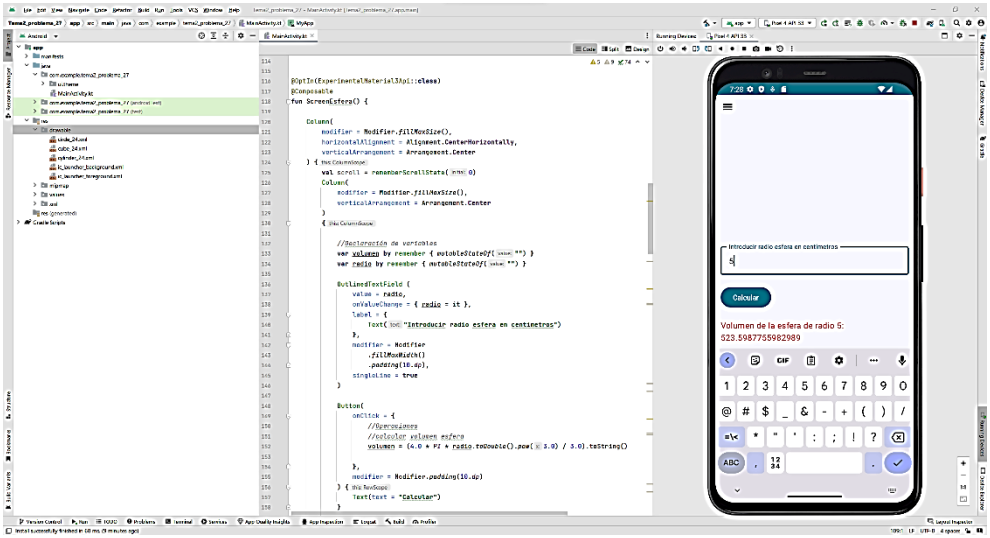
```
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_27Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    MyApp()
                }
            }
        }
    }
}

data class TabInfo(val title: String, val icon: Painter)


@Composable
fun MyApp() {
    var selectedTabIndex by remember { mutableStateOf(0) }
    //val tabs = listOf("Screen 1", "Screen 2", "Screen 3")
    var expanded by remember { mutableStateOf(false) }


    val tabs = listOf(
        TabInfo("Esfera", painterResource(id = R.drawable.circle_24)),
        TabInfo("Cilindro", painterResource(id = R.drawable.cylinder_24)),
        TabInfo("Cubo", painterResource(id = R.drawable.cube_24))
    )


    when (selectedTabIndex) {
        0 -> ScreenEsfera()
        1 -> ScreenCilindro()
        2 -> ScreenCubo()
    }

    Box(
        modifier = Modifier
```

```
            .fillMaxSize()
            .wrapContentSize(Alignment.TopStart)
    ) {
        IconButton(onClick = { expanded = true }) {
            Icon(Icons.Default.Menu, contentDescription = "Localized description")
        }

        DropdownMenu(
            expanded = expanded,
            onDismissRequest = { expanded = false }
        ) {
            tabs.forEachIndexed { index, tabInfo ->
                DropdownMenuItem(
                    text = { Text(text = tabInfo.title) },
                    onClick = { selectedTabIndex = index },
                    leadingIcon = {
                        Icon(
                            painter = tabInfo.icon,
                            contentDescription = null // Opcional: Puedes agregar una descripción de
contenido aquí
                        )
                    })
            }

        }//DropdownMenu


    }
}
```

Paso 3:

Codificar los cálculos de las diferentes figuras geométricas.

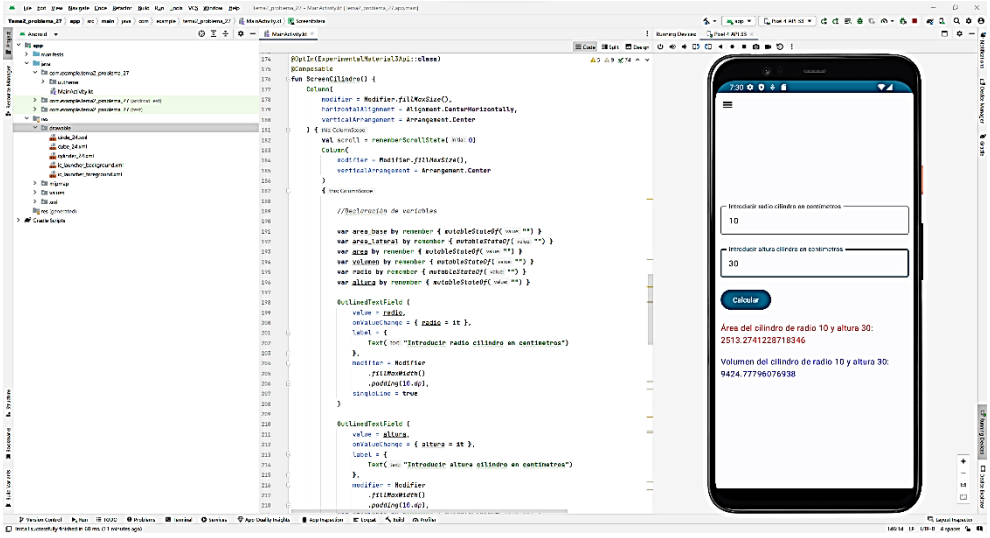
Esfera:




```
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun ScreenEsfera() {

    Column(
        modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        val scroll = rememberScrollState(0)
        Column(
```

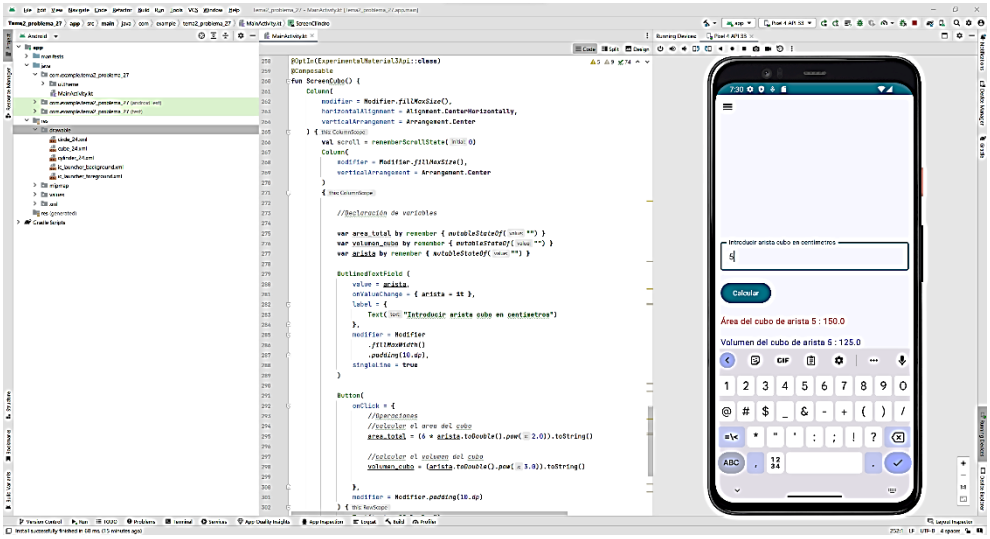
```
            modifier = Modifier.fillMaxSize(),
            verticalArrangement = Arrangement.Center
        )
        {

            //Declaración de variables
            var volumen by remember { mutableStateOf("") }
            var radio by remember { mutableStateOf("") }

            OutlinedTextField (
                value = radio,
                onValueChange = { radio = it },
                label = {
                    Text("Introducir radio esfera en centímetros")
                },
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(10.dp),
                singleLine = true
            )

            Button(
                onClick = {
                    //Operaciones
                    //calcular volumen esfera
                    volumen = (4.0 * PI * radio.toDouble().pow(3.0) / 3.0).toString()

                },
                modifier = Modifier.padding(10.dp)
            ) {
                Text(text = "Calcular")
            }


            //Mostrar por pantalla

            Text(
                text = "Volumen de la esfera de radio $radio: "+volumen,
                color= Color.Red,
                modifier = Modifier.padding(10.dp)
            )

        }

    }
}
```

Cilindro



```
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun ScreenCilindro() {
    Column(
        modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        val scroll = rememberScrollState(0)
        Column(
            modifier = Modifier.fillMaxSize(),
            verticalArrangement = Arrangement.Center
        )
        {

            //Declaración de variables

            var area_base by remember { mutableStateOf("") }
            var area_lateral by remember { mutableStateOf("") }
            var area by remember { mutableStateOf("") }
            var volumen by remember { mutableStateOf("") }
            var radio by remember { mutableStateOf("") }
            var altura by remember { mutableStateOf("") }

            OutlinedTextField (
                value = radio,
                onValueChange = { radio = it },
                label = {
                    Text("Introducir radio cilindro en centímetros")
                },
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(10.dp),
                singleLine = true
            )

            OutlinedTextField (
                value = altura,
                onValueChange = { altura = it },
                label = {
                    Text("Introducir altura cilindro en centímetros")
                },
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(10.dp),
                singleLine = true
            )


            Button(
                onClick = {
                    //Operaciones
                    //calcular el area del cilindro
                    area_base = (2 * PI * radio.toDouble().pow(2.0)).toString()
                    area_lateral = (2 * PI * radio.toDouble() * altura.toDouble()).toString()
                    area = (area_base.toDouble() + area_lateral.toDouble()).toString()

                    //calcular el volumen del cilindro
                    volumen = (PI * radio.toDouble().pow(2.0) * altura.toDouble()).toString()

                },
                modifier = Modifier.padding(10.dp)
            ) {
                Text(text = "Calcular")
            }


            //Mostrar por pantalla
            Text(
                text ="Área del cilindro de radio $radio y altura $altura: "+area,
                color= Color.Red,
                modifier = Modifier.padding(10.dp)
            )
            Text(
                text ="Volumen del cilindro de radio $radio y altura $altura: "+volumen,
                color= Color.Blue,
                modifier = Modifier.padding(10.dp)
            )

        }

    }
}
```

Cubo



```
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun ScreenCubo() {
    Column(
        modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        val scroll = rememberScrollState(0)
        Column(
            modifier = Modifier.fillMaxSize(),
            verticalArrangement = Arrangement.Center
        )
        {

            //Declaración de variables

            var area_total by remember { mutableStateOf("") }
            var volumen_cubo by remember { mutableStateOf("") }
            var arista by remember { mutableStateOf("") }

            OutlinedTextField (
                value = arista,
                onValueChange = { arista = it },
                label = {
                    Text("Introducir arista cubo en centímetros")
                },
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(10.dp),
                singleLine = true
            )

            Button(
                onClick = {
                    //Operaciones
                    //calcular el area del cubo
                    area_total = (6 * arista.toDouble().pow(2.0)).toString()

                    //calcular el volumen del cubo
                    volumen_cubo = (arista.toDouble().pow(3.0)).toString()

                },
                modifier = Modifier.padding(10.dp)
            ) {
                Text(text = "Calcular")
```

```
            }

            //Mostrar por pantalla
            Text(
                text ="Área del cubo de arista $arista : "+area_total,
                color= Color.Red,
                modifier = Modifier.padding(10.dp)
            )
            Text(
                text ="Volumen del cubo de arista $arista : "+volumen_cubo,
                color= Color.Blue,
                modifier = Modifier.padding(10.dp)
            )

        }

    }
}
```

Código completo

MainActivity.kt:

```
package com.example.tema2_problema_27

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.layout.wrapContentSize
import androidx.compose.foundation.rememberScrollState
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.Menu
import androidx.compose.material3.Button
import androidx.compose.material3.DropdownMenu
import androidx.compose.material3.DropdownMenuItem
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.Icon
import androidx.compose.material3.IconButton
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.graphics.painter.Painter
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.tema2_problema_27.ui.theme.Tema2_problema_27Theme
import kotlin.math.PI
import kotlin.math.pow

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Tema2_problema_27Theme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    MyApp()
                }
            }
```

```
        }
    }
}

data class TabInfo(val title: String, val icon: Painter)


@Composable
fun MyApp() {
    var selectedTabIndex by remember { mutableStateOf(0) }
    //val tabs = listOf("Screen 1", "Screen 2", "Screen 3")
    var expanded by remember { mutableStateOf(false) }


    val tabs = listOf(
        TabInfo("Esfera", painterResource(id = R.drawable.circle_24)),
        TabInfo("Cilindro", painterResource(id = R.drawable.cylinder_24)),
        TabInfo("Cubo", painterResource(id = R.drawable.cube_24))
    )


    when (selectedTabIndex) {
        0 -> ScreenEsfera()
        1 -> ScreenCilindro()
        2 -> ScreenCubo()
    }

    Box(
        modifier = Modifier
            .fillMaxSize()
            .wrapContentSize(Alignment.TopStart)
    ) {
        IconButton(onClick = { expanded = true }) {
            Icon(Icons.Default.Menu, contentDescription = "Localized description")
        }

        DropdownMenu(
            expanded = expanded,
            onDismissRequest = { expanded = false }
        ) {
            tabs.forEachIndexed { index, tabInfo ->
                DropdownMenuItem(
                    text = { Text(text = tabInfo.title) },
                    onClick = { selectedTabIndex = index },
                    leadingIcon = {
                        Icon(
                            painter = tabInfo.icon,
                            contentDescription = null // Opcional: Puedes agregar una descripción de
contenido aquí
                        )
                    })
            }

        }//DropdownMenu


    }
}



@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun ScreenEsfera() {

    Column(
        modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        val scroll = rememberScrollState(0)
        Column(
            modifier = Modifier.fillMaxSize(),
            verticalArrangement = Arrangement.Center
        )
        {

            //Declaración de variables
            var volumen by remember { mutableStateOf("") }
            var radio by remember { mutableStateOf("") }

            OutlinedTextField (
```

```
                value = radio,
                onValueChange = { radio = it },
                label = {
                    Text("Introducir radio esfera en centímetros")
                },
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(10.dp),
                singleLine = true
            )

            Button(
                onClick = {
                    //Operaciones
                    //calcular volumen esfera
                    volumen = (4.0 * PI * radio.toDouble().pow(3.0) / 3.0).toString()

                },
                modifier = Modifier.padding(10.dp)
            ) {
                Text(text = "Calcular")
            }


            //Mostrar por pantalla

            Text(
                text = "Volumen de la esfera de radio $radio: "+volumen,
                color= Color.Red,
                modifier = Modifier.padding(10.dp)
            )

        }

    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun ScreenCilindro() {
    Column(
        modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        val scroll = rememberScrollState(0)
        Column(
            modifier = Modifier.fillMaxSize(),
            verticalArrangement = Arrangement.Center
        )
        {

            //Declaración de variables

            var area_base by remember { mutableStateOf("") }
            var area_lateral by remember { mutableStateOf("") }
            var area by remember { mutableStateOf("") }
            var volumen by remember { mutableStateOf("") }
            var radio by remember { mutableStateOf("") }
            var altura by remember { mutableStateOf("") }

            OutlinedTextField (
                value = radio,
                onValueChange = { radio = it },
                label = {
                    Text("Introducir radio cilindro en centímetros")
                },
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(10.dp),
                singleLine = true
            )

            OutlinedTextField (
                value = altura,
                onValueChange = { altura = it },
                label = {
                    Text("Introducir altura cilindro en centímetros")
                },
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(10.dp),
```

```
                singleLine = true
            )


            Button(
                onClick = {
                    //Operaciones
                    //calcular el area del cilindro
                    area_base = (2 * PI * radio.toDouble().pow(2.0)).toString()
                    area_lateral = (2 * PI * radio.toDouble() * altura.toDouble()).toString()
                    area = (area_base.toDouble() + area_lateral.toDouble()).toString()

                    //calcular el volumen del cilindro
                    volumen = (PI * radio.toDouble().pow(2.0) * altura.toDouble()).toString()

                },
                modifier = Modifier.padding(10.dp)
            ) {
                Text(text = "Calcular")
            }


            //Mostrar por pantalla
            Text(
                text ="Área del cilindro de radio $radio y altura $altura: "+area,
                color= Color.Red,
                modifier = Modifier.padding(10.dp)
            )
            Text(
                text ="Volumen del cilindro de radio $radio y altura $altura: "+volumen,
                color= Color.Blue,
                modifier = Modifier.padding(10.dp)
            )

        }

    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun ScreenCubo() {
    Column(
        modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        val scroll = rememberScrollState(0)
        Column(
            modifier = Modifier.fillMaxSize(),
            verticalArrangement = Arrangement.Center
        )
        {

            //Declaración de variables

            var area_total by remember { mutableStateOf("") }
            var volumen_cubo by remember { mutableStateOf("") }
            var arista by remember { mutableStateOf("") }

            OutlinedTextField (
                value = arista,
                onValueChange = { arista = it },
                label = {
                    Text("Introducir arista cubo en centímetros")
                },
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(10.dp),
                singleLine = true
            )

            Button(
                onClick = {
                    //Operaciones
                    //calcular el area del cubo
                    area_total = (6 * arista.toDouble().pow(2.0)).toString()

                    //calcular el volumen del cubo
                    volumen_cubo = (arista.toDouble().pow(3.0)).toString()

                },
```

```
                modifier = Modifier.padding(10.dp)
            ) {
                Text(text = "Calcular")
            }

            //Mostrar por pantalla
            Text(
                text ="Área del cubo de arista $arista : "+area_total,
                color= Color.Red,
                modifier = Modifier.padding(10.dp)
            )
            Text(
                text ="Volumen del cubo de arista $arista : "+volumen_cubo,
                color= Color.Blue,
                modifier = Modifier.padding(10.dp)
            )

        }

    }
}
```