



SEMANA 2 / Actividad N°1

Nombre de la Unidad:	FUNDAMENTOS DE ANDROID
Nombre de la Asignatura:	PROGRAMACIÓN MOBILE

Aprendizaje Esperado

- Comprender la sintaxis y estructura básica de un programa Kotlin
- Comprobar algunos conceptos básicos del lenguaje de programación Kotlin
- Utilizar Jetpack Compose y Vistas XML en un mismo proyecto

Respuestas Esperadas

A continuación, encontrará el desarrollo de los ejercicios que resolvió en la semana. Contraste las respuestas entregadas por el docente con las desarrolladas por usted. En caso de que no coincidan, y persistan dudas, le invitamos a repasar los contenidos y/ o consultar con su profesor.

Para el caso de los ejercicios de selección múltiple solo encontrará la mitad de las respuestas de las preguntas inicialmente planteadas en los ejercicios. Si tiene alguna duda con las respuestas que no se incluyen, debe comunicarse con su docente para resolverlo.

ÍTEM I: Selección múltiple

Marque con una X la respuesta correcta.

¿Cuál es la forma correcta de declarar una variable mutable en Kotlin?

- a) val variable: Int = 10
- b) var variable: Int = 10
- c) let variable: Int = 10

Respuesta correcta: b)

¿Cuál es la forma correcta de declarar una constante en Kotlin?

- a) const val PI: Double = 3.14159
- b) final val PI: Double = 3.14159
- c) val PI: Double = 3.14159

Respuesta correcta: c)

¿Cuál es la forma correcta de declarar una clase en Kotlin?

- a) class MyClass {}
- b) Class MyClass {}
- c) def MyClass {}

Respuesta correcta: a)

I. ÍTEM II: Desarrollo

Se solicita crear una aplicación para Android de cálculo del perímetro de un rectángulo y triángulo.

Requisitos:

1. Diseñe una jerarquía de clases usando herencia en Kotlin para solucionar la problemática del caso (figuras geométricas)
2. Desarrolle la pantalla de inicio donde podrá navegar hacia las otras pantallas usando la tecnología de su elección (Jetpack Compose o Vistas XML)
3. Desarrolle la pantalla para el cálculo del perímetro del rectángulo usando Jetpack Compose
4. Desarrolle la pantalla para el cálculo del perímetro del triángulo usando Vistas XML
5. Coloque un botón para volver atrás en cada pantalla
6. Gestione a través de evento onClick sobre los botones correspondientes para mostrar los resultados de los perímetros

```
abstract class Figura {  
    abstract fun calcularPerimetro(): Int  
}  
  
class Rectangulo(val base:Int, val altura:Int) : Figura() {  
    override fun calcularPerimetro(): Int = (base + altura) * 2  
}  
  
class Triangulo(val ladoA:Int, val ladoB:Int, val ladoC:Int) : Figura() {  
    override fun calcularPerimetro(): Int = ladoA + ladoB + ladoC  
}
```

Código 1 Jerarquía de clases de figuras geométricas

```
class MainActivity : ComponentActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContent {  
            PantallaInicio()  
        }  
    }  
}  
  
@Preview  
@Composable  
fun PantallaInicio() {  
    val contexto = LocalContext.current as ComponentActivity  
  
    Column(  
        modifier = Modifier.fillMaxSize(),  
        verticalArrangement = Arrangement.Center,  
        horizontalAlignment = Alignment.CenterHorizontally  
    ) {  
        Text(  

```

```

        text = "Perímetro Figuras",
        style = TextStyle(
            fontSize = 30.sp,
            fontWeight = FontWeight.Bold
        )
    )
    Spacer(modifier = Modifier.height(30.dp))

    Button(modifier = Modifier.height(60.dp).width(150.dp), onClick = {
        val intent: Intent = Intent(contexto, PerimetroRectanguloActivity::class.java)
        contexto.startActivity(intent)
    }) {
        Text("Rectángulo")
    }
    Spacer(modifier = Modifier.height(10.dp))
    Button(modifier = Modifier.height(60.dp).width(150.dp), onClick = {
        val intent: Intent = Intent(contexto, PerimetroTrianguloActivity::class.java)
        contexto.startActivity(intent)
    }) {
        Text("Triángulo")
    }
}

```

Código 2 MainActivity usando Jetpack Compose - Pantalla Inicio

```

class PerimetroRectanguloActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            PantallaRectangulo()
        }
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Preview
@Composable
fun PantallaRectangulo() {
    var base by remember { mutableStateOf("") }
    var altura by remember { mutableStateOf("") }
    var resultado by remember { mutableStateOf("") }
    val contexto = LocalContext.current as ComponentActivity

    Column(
        modifier=Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text(
            text = "Perímetro Rectángulo",
            style = TextStyle(fontSize = 30.sp, fontWeight = FontWeight.Bold)
        )
        Spacer(modifier = Modifier.height(20.dp))
        TextField(
            placeholder = {Text("Base")},

```

```

        value = base,
        onChange = {base = it},
        keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Number)
    )
    TextField(
        placeholder = {Text("Altura")},
        value = altura,
        onChange = {altura = it},
        keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Number)
    )
    Spacer(modifier = Modifier.height(20.dp))
    Button(onClick = {
        val b = base.toIntOrNull() ?: 0
        val a = altura.toIntOrNull() ?: 0
        val perimetro = Rectangulo(b, a).calcularPerimetro()
        resultado = "El perímetro es de ${perimetro}"
    }) {
        Text("Calcular Perímetro")
    }
    Spacer(modifier = Modifier.height(20.dp))
    Text(resultado)
    Spacer(modifier = Modifier.height(20.dp))
    Button(onClick = { contexto.finish() }) {
        Text("<- Volver")
    }
}

```

Código 3 Activity para Rectángulo con Jetpack Compose

```

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

        val btnCalcular = findViewById<Button>(R.id.btnCalcular)
        btnCalcular.setOnClickListener {
            val resultado = findViewById<TextView>(R.id.tvResultado)

            val ladoA = findViewById<EditText>(R.id.etLadoA)
            val ladoB = findViewById<EditText>(R.id.etLadoB)
            val ladoC = findViewById<EditText>(R.id.etLadoC)

            val a = ladoA.text.toString().toIntOrNull() ?: 0
            val b = ladoB.text.toString().toIntOrNull() ?: 0
            val c = ladoC.text.toString().toIntOrNull() ?: 0

            val triangulo = Triangulo(a, b, c)
            val perimetro = triangulo.calcularPerimetro()
            resultado.text = "El perímetro es ${perimetro}"
        }
    }

    fun volverAtras(view: View) {
        finish()
    }
}

```

```
}  
}
```

Código 4 Activity para Triángulo usando Vista XML

```
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:app="http://schemas.android.com/apk/res-auto"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout_width="match_parent"  
android:layout_height="match_parent"  
android:padding="10dp"  
tools:context=".PerimetroTrianguloActivity">  
  
    <TextView  
        android:id="@+id/tvTitulo"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Perímetro Triángulo"  
        android:textSize="30dp"  
        android:textStyle="bold"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintTop_toTopOf="parent"  
        app:layout_constraintVertical_bias="0.2" />  
  
    <EditText  
        android:id="@+id/etLadoA"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_marginTop="32dp"  
        android:ems="10"  
        android:hint="Lado A"  
        android:inputType="number"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintTop_toBottomOf="@id/tvTitulo" />  
  
    <EditText  
        android:id="@+id/etLadoB"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_marginTop="16dp"  
        android:ems="10"  
        android:hint="Lado B"  
        android:inputType="number"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintTop_toBottomOf="@id/etLadoA" /> />  
  
    <EditText  
        android:id="@+id/etLadoC"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_marginTop="16dp"
```

```

    android:ems="10"
    android:hint="Lado C"
    android:inputType="number"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/etLadoB" />

<Button
    android:id="@+id/btnCalcular"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:text="Calcular"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etLadoC" />

<TextView
    android:id="@+id/tvResultado"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text=""
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btnCalcular" />

<Button
    android:id="@+id/btnVolver"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="&lt;- Volver"
    android:onClick="volverAtras"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Código 5 Vista XML de la pantalla para el Triángulo usando ConstraintLayout

Para que funcione la vista XML debemos configurar un Theme de compatibilidad en Android Manifest.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.S02PerimetroFiguras"
        tools:targetApi="31">
        <activity

```



```
    android:name=".PerimetroTrianguloActivity"
    android:theme="@style/Theme.AppCompat.Light"
    android:exported="false" />
<activity
    android:name=".PerimetroRectanguloActivity"
    android:exported="false"
    android:label="@string/title_activity_perimetro_rectangulo"
    android:theme="@style/Theme.S02PerimetroFiguras" />
<activity
    android:name=".MainActivity"
    android:exported="true"
    android:label="@string/app_name"
    android:theme="@style/Theme.S02PerimetroFiguras">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
</application>
</manifest>
```

Código 6 Archivo AndroidManifest.xml

