

ASIGNATURA: Desarrollo de App Móviles.

TEMA: Configuración de entorno de desarrollo Kotlin MultiPlataforma

ACTIVIDAD DE ENSEÑANZA – APRENDIZAJE – EVALUACIÓN: Configuración e implementación de proyectos con Android y Kotlin Multiplataforma.

TIEMPO DE LA ACTIVIDAD DE E-A-E: 2 horas

TIEMPO DE LA GUIA DE APRENDIZAJE: 2 horas

ORIENTACIONES GENERALES:

- Seguir la siguiente guía para la configuración de Kotlin Multiplataforma en la herramienta Android.

Concepto.

Kotlin Multiplataforma (KMP) es una tecnología que permite crear aplicaciones para múltiples plataformas, como **Android**, **iOS**, **escritorio**, **web** y **servidor**. KMP es de código abierto y fue desarrollada por JetBrains.

Para preparar Kotlin Multi plataforma para iniciar el proceso de desarrollo siga los siguientes pasos:

1. Abra la aplicación Android, luego pulse sobre la opción **Main Menu**, y luego elija y pulse la opción **Settings** como se muestra en las figuras 1 y 2.

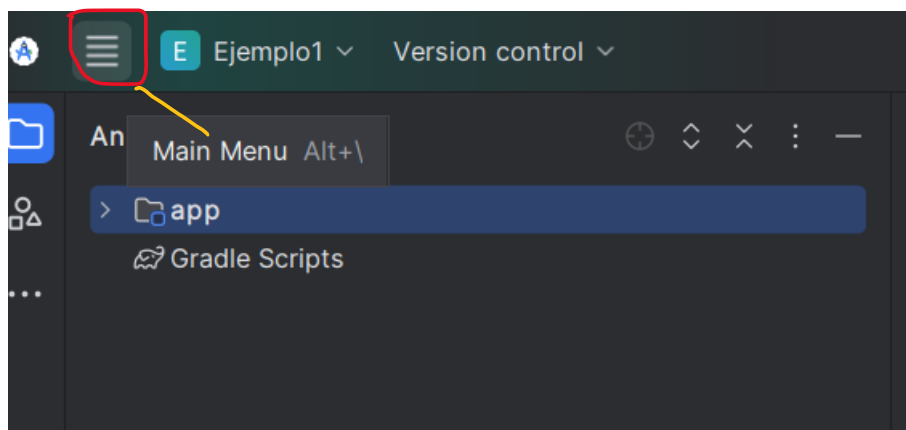


Fig. 1

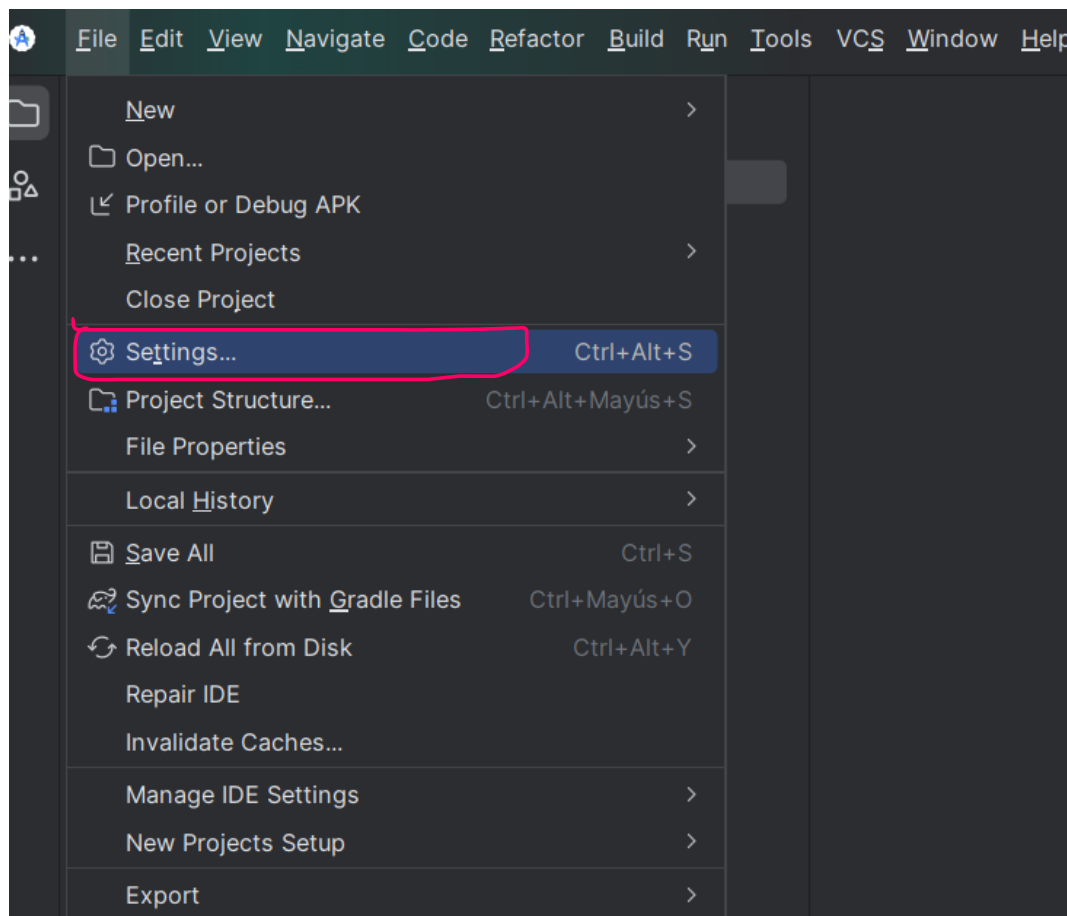


Fig. 2

- Después de haber pulsado sobre la opción **Settings**, escriba en el campo de búsqueda donde aparece la imagen de “**Lupa**”, la palabra “**plugin**”, para que aparezca dicha opción, pulse sobre ella y luego sobre el campo de plugin escriba la palabra “**Kotlin**” posteriormente instale los dos plugin (**Kotlin** y **Kotlin Multiplatform**) y por último en cada caso pulse sobre el botón “**Ok**” para proceder a su instalación. Ver figura 3.

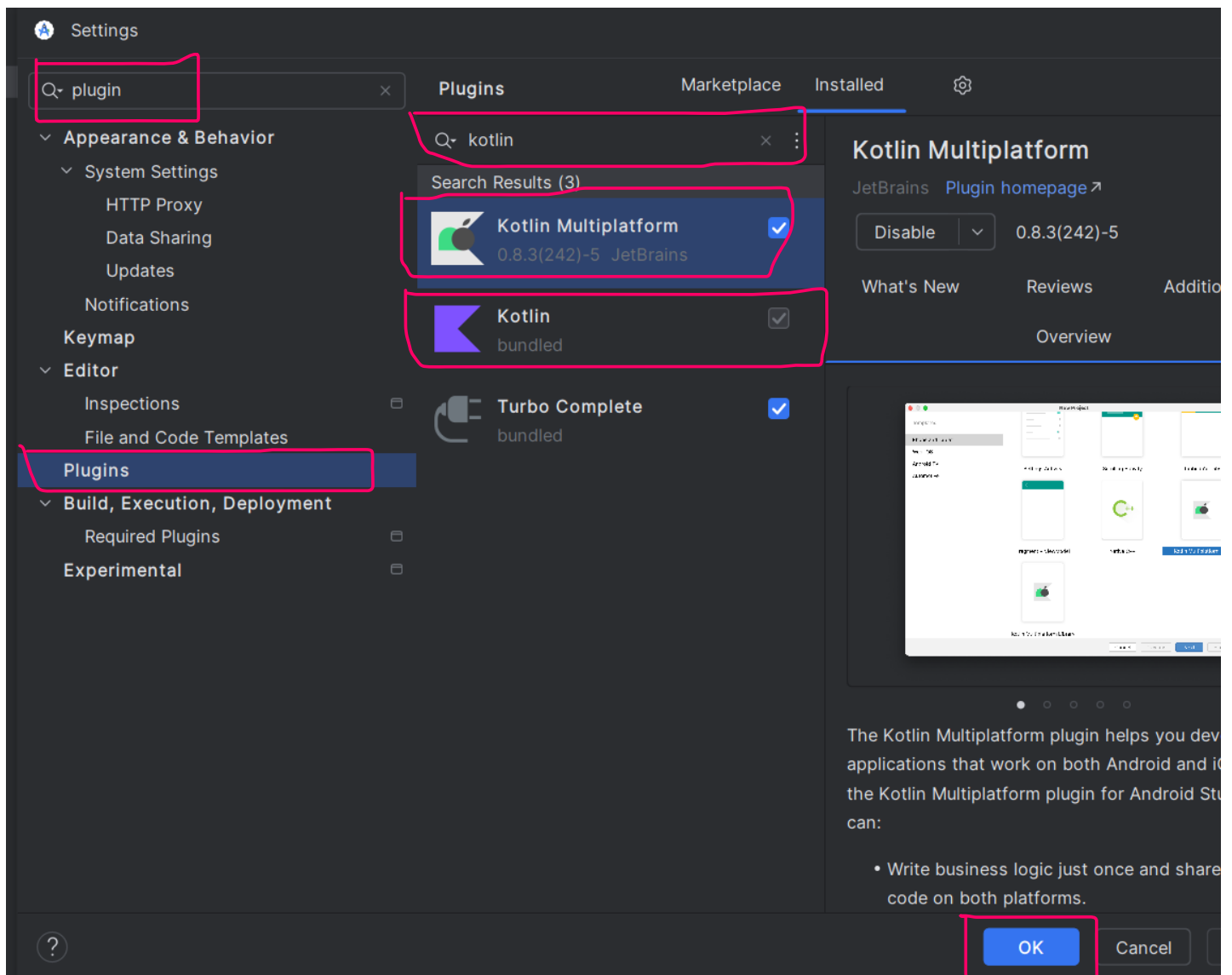
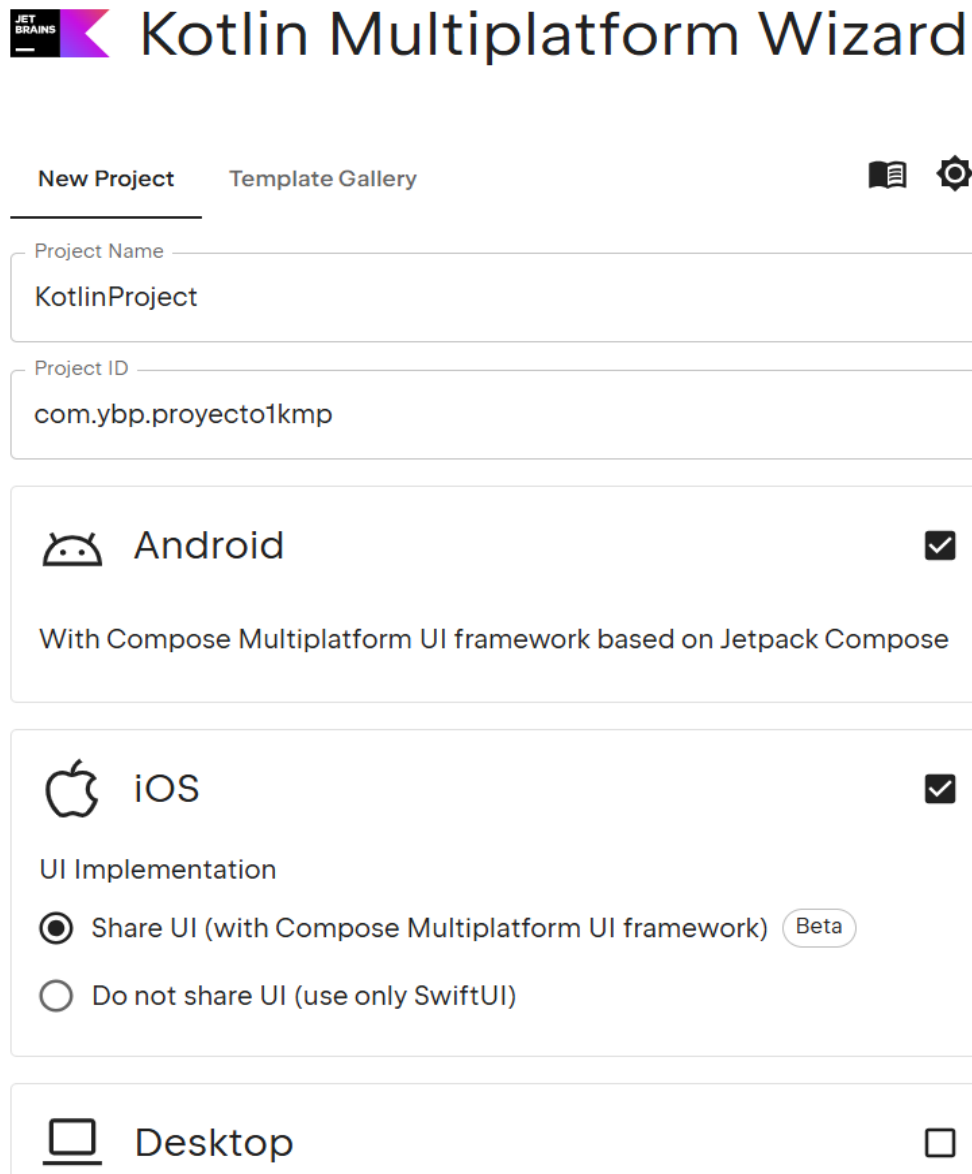




Fig. 3

Con los procedimientos anteriores ya está listo y configurado el entorno de desarrollo

3. Para crear un proyecto abra un visualizador y escriba la siguiente URL “ <https://kmp.jetbrains.com/> ” posteriormente, aparecerá un formulario desde donde podrá crear el proyecto seleccionando la o las diferentes plataformas de programación y ejecución del código. Ver figura 4.




Kotlin Multiplatform Wizard


New Project **Template Gallery**  

Project Name

Project ID

 **Android** ☒

With Compose Multiplatform UI framework based on Jetpack Compose

 **iOS** ☒

UI Implementation

☒ Share UI (with Compose Multiplatform UI framework) Beta

☐ Do not share UI (use only SwiftUI)


 **Desktop** ☐

Fig. 4

4. Después de haber configurado el proyecto **Kotlin Multiplatform**, descargue el proyecto pulsando sobre el botón **DOWNLOAD**. Ver figura 5.

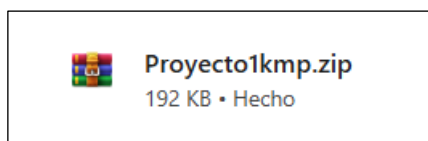


Fig. 5

5. Luego de haber descargado el proyecto ubíquelo en un lugar deseado del PC y luego descomprímalo. Ver figura 6.

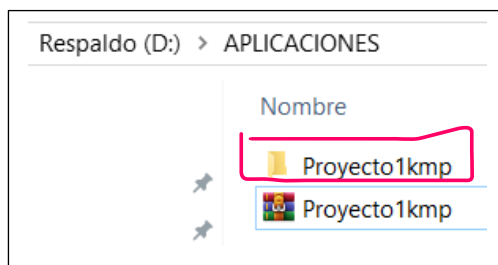


Fig. 6

6. Abra el proyecto desde Android, seleccione la opción **File**, luego **Open**, por último, pulse sobre el botón **Ok**. Ver figuras 7 y 8.

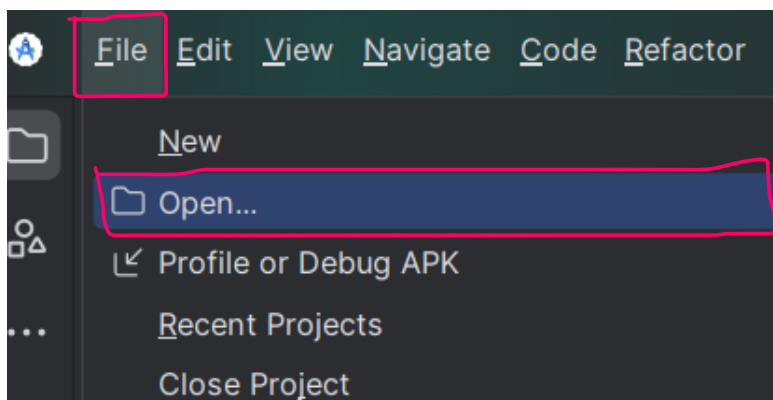


Fig. 7

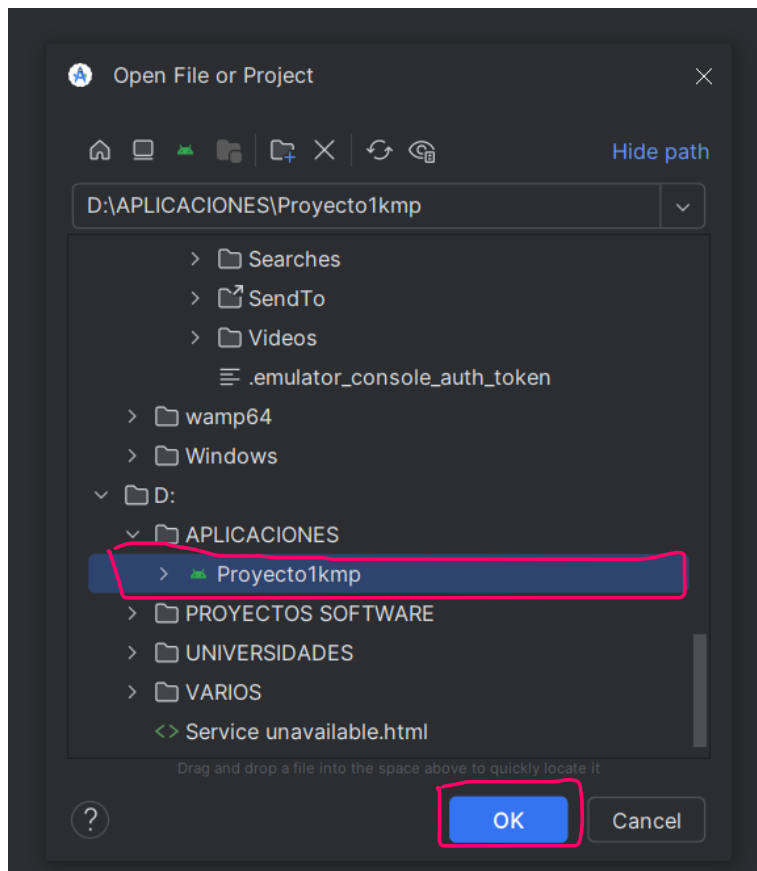


Fig. 8

7. Después de pulsar sobre el botón Ok, seleccione la ventana donde se abrirá el proyecto y luego se creará. Ver figuras 9 y 10.

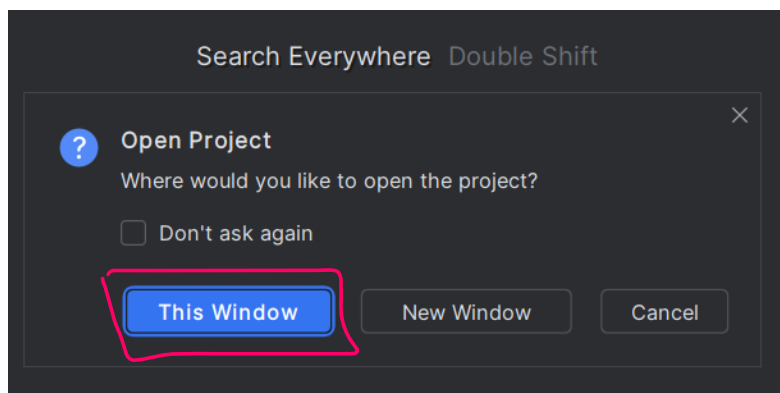


Fig. 9

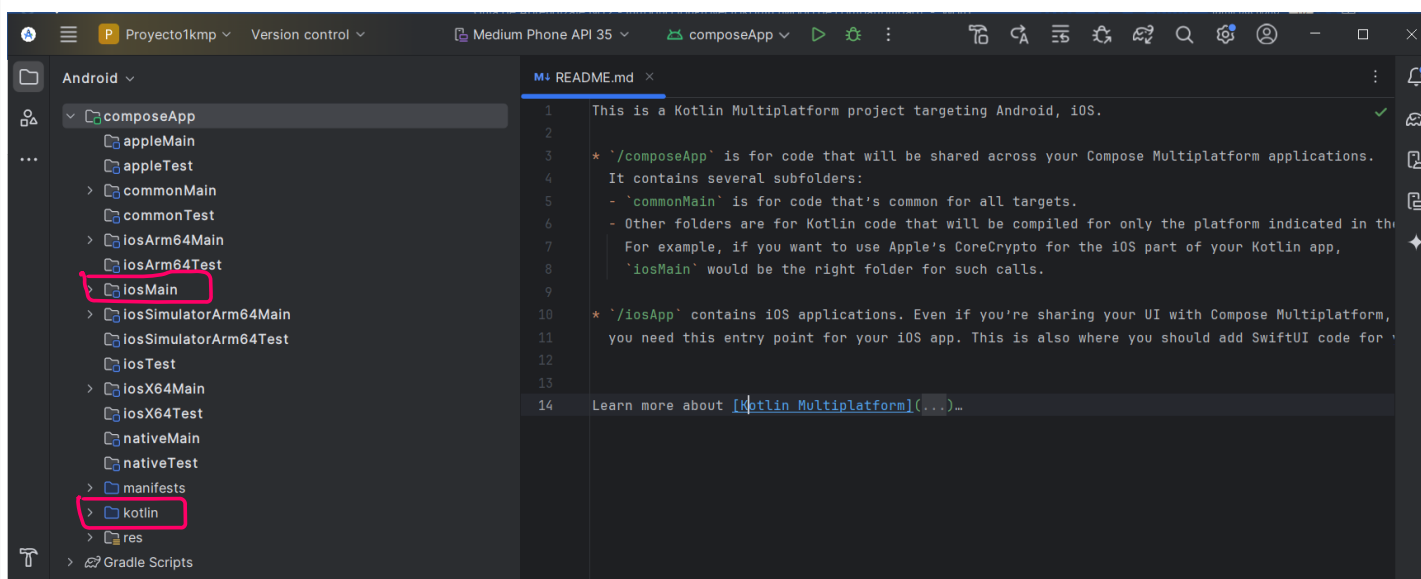


Fig. 10

8. Después de haber creado el proyecto cambie la vista creada de la pestaña **Android** a la vista **Project**. Para lo cual pulse sobre la opción **Android** y posteriormente seleccione la opción **Project**. Ver figura 11y 12.

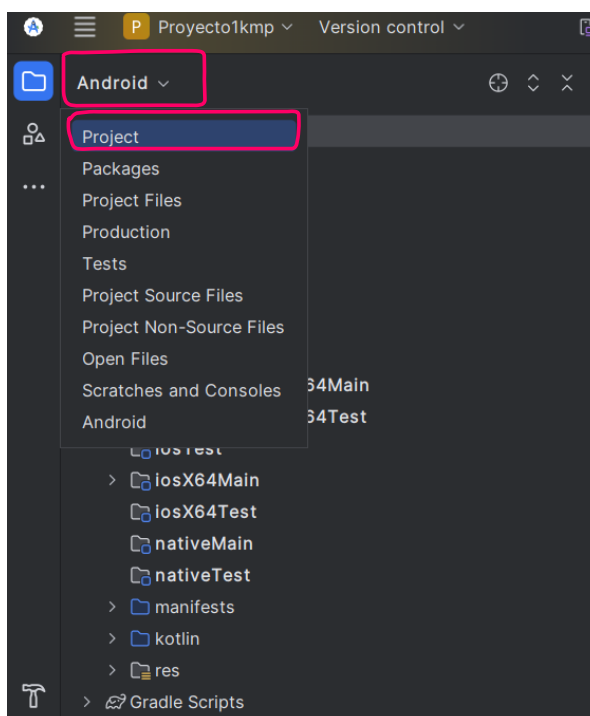


Fig. 11

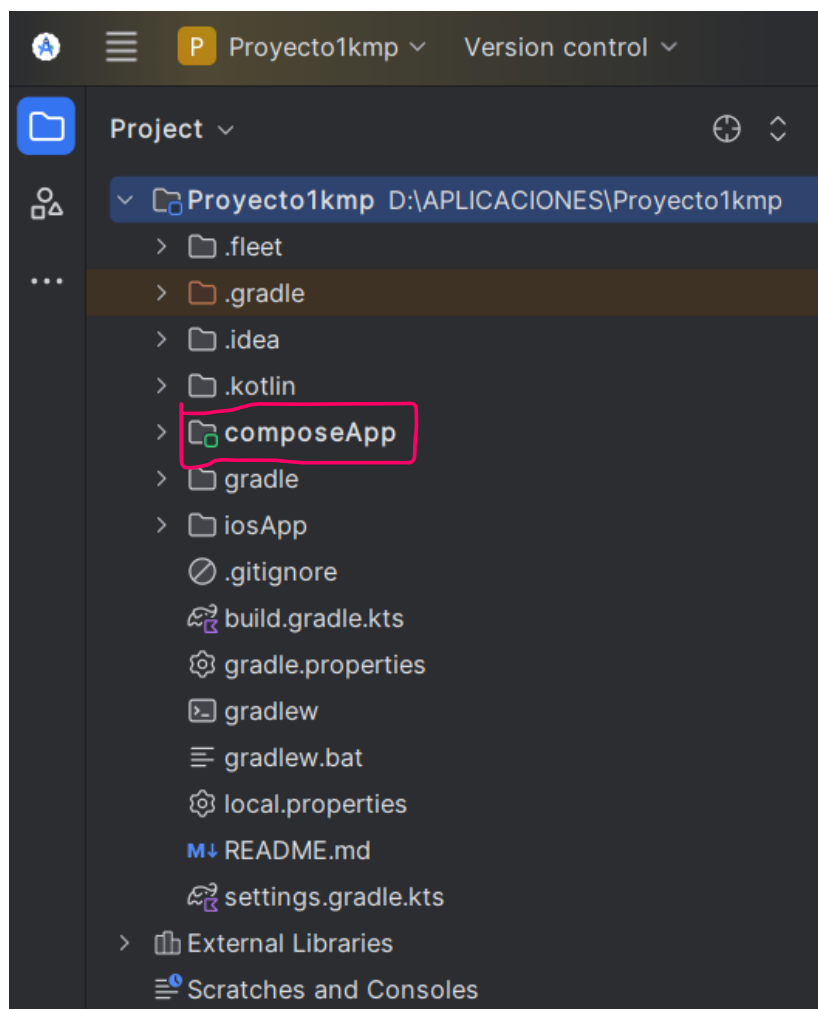


Fig. 12

9. Despliegue el paquete **composeApp** y luego el paquete **SRC**, donde encontrará las diferentes plataformas seleccionadas y a publicar. Ver figura 13.

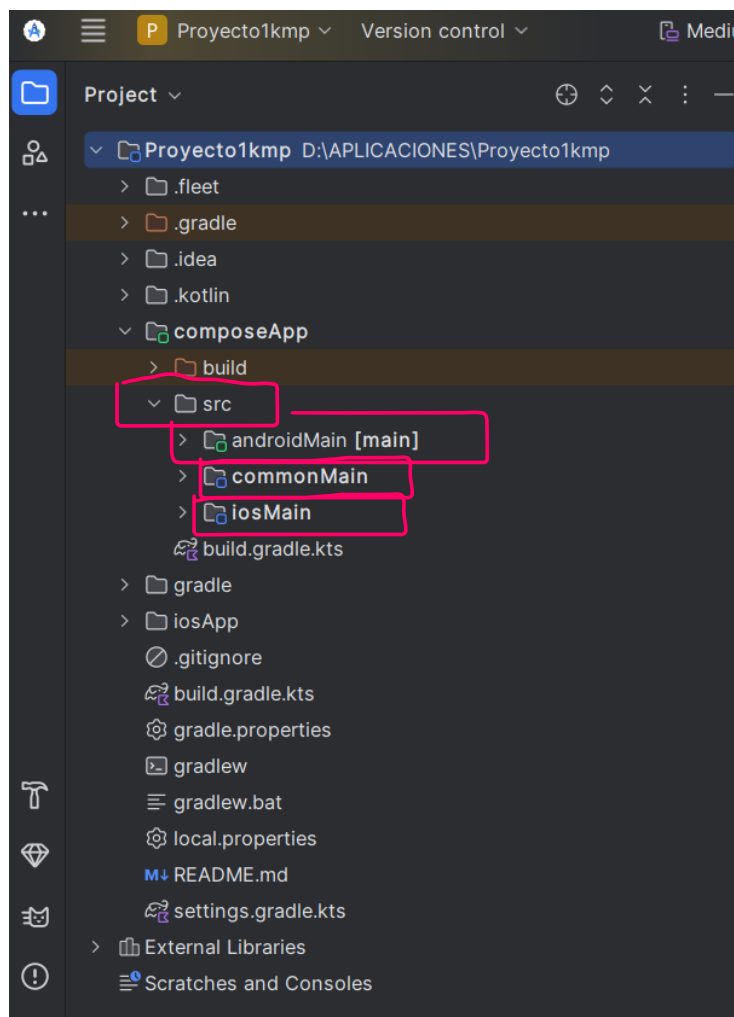


Fig. 13

- La carpeta **androidMain**, contiene la codificación para publicar en Android.
- La carpeta **commonMain**, es la carpeta genérica donde se escribe el código de programa de las App, el cual será común para las plataformas seleccionadas.
- La carpeta **iosMain**, contiene la codificación para publicar en IOS.

10. Posteriormente, ejecute la App, para lo cual pulse sobre el botón que representa el símbolo **Play**. Ver figuras 14 y 15.

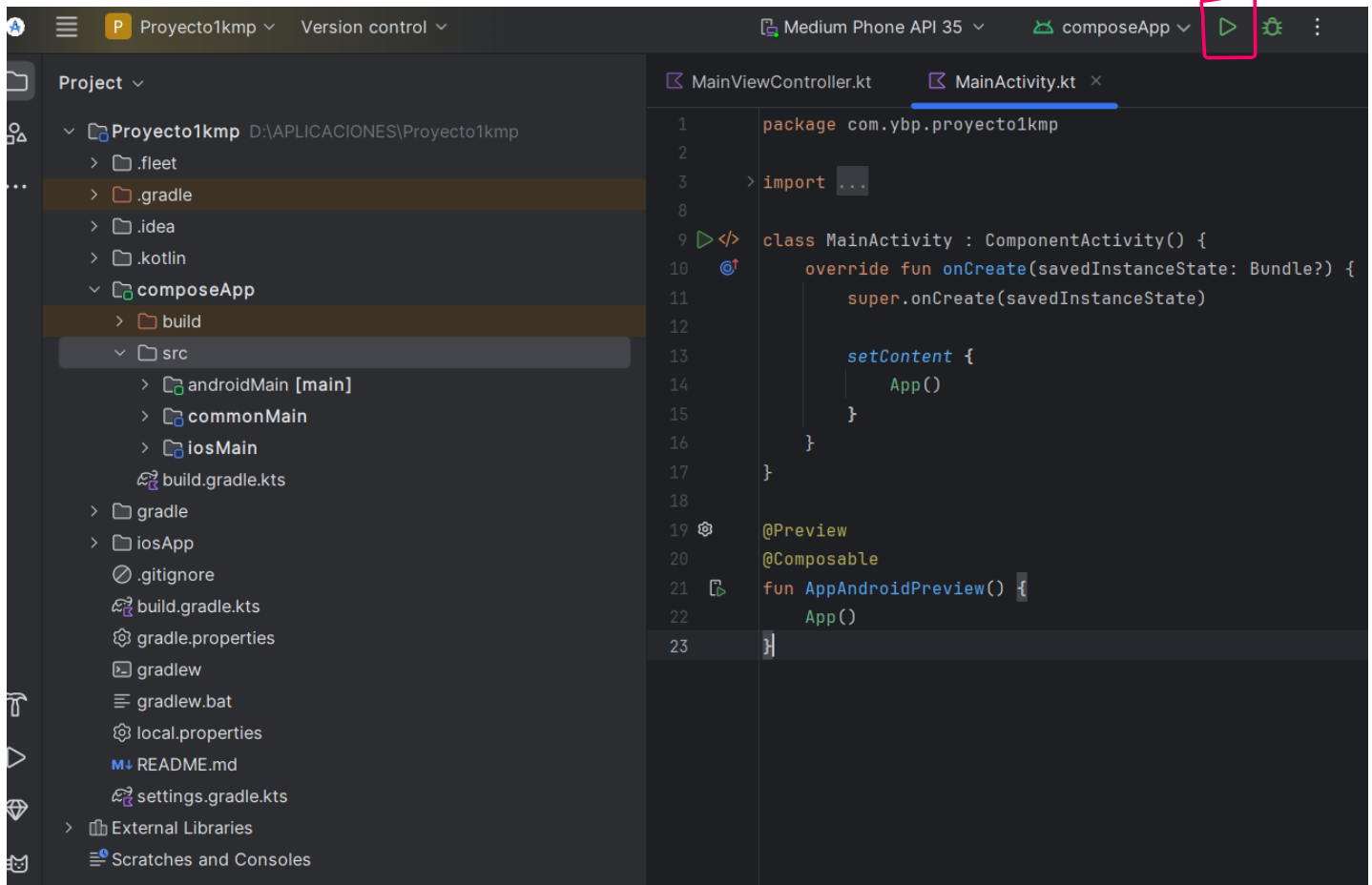


Fig. 14

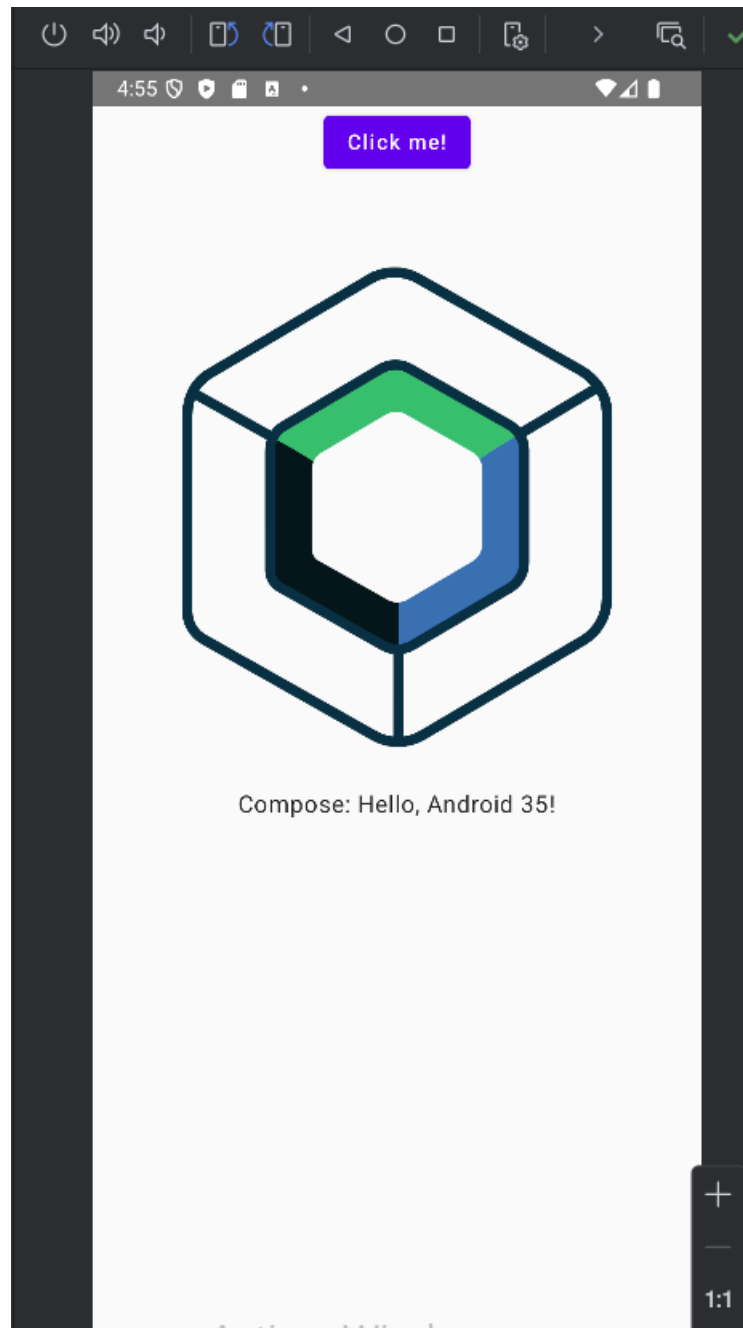
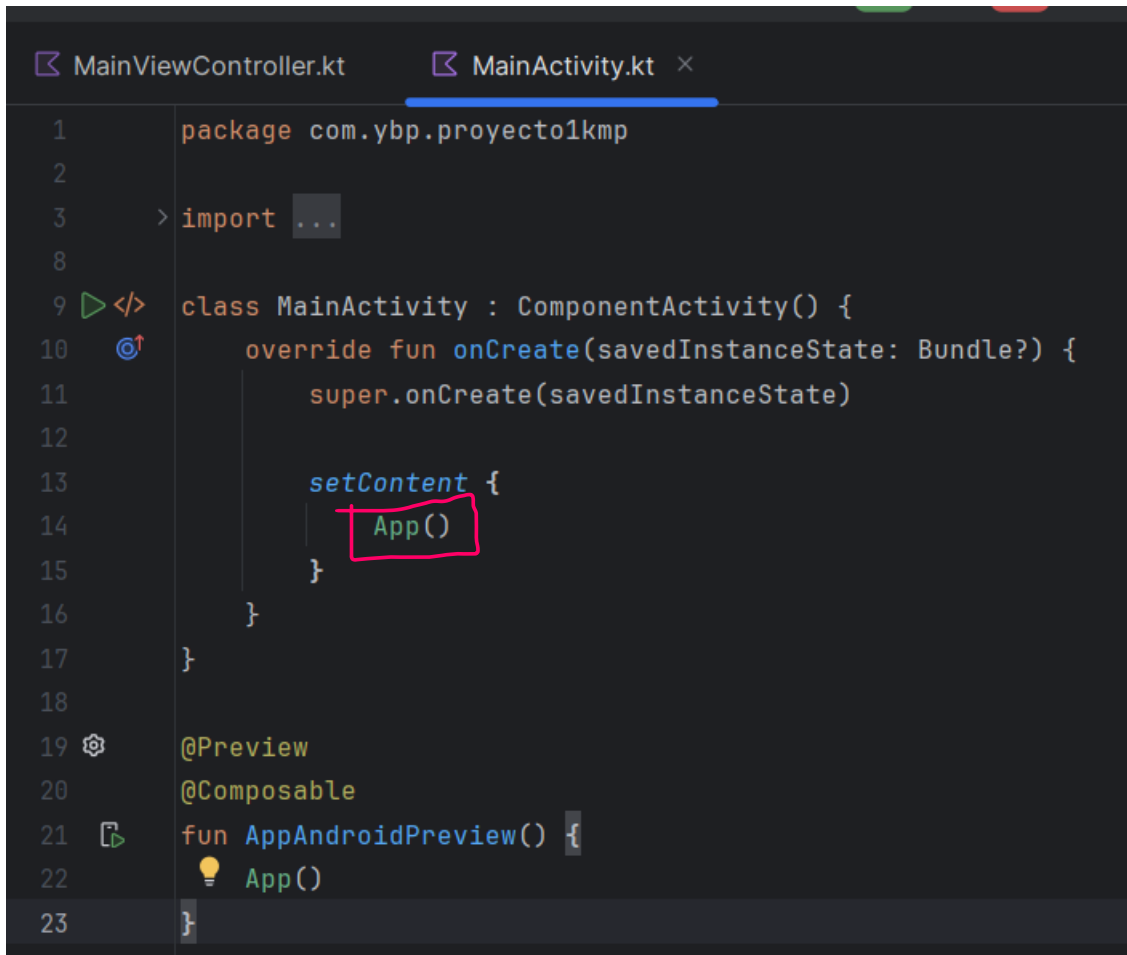


Fig. 15

Preparación y Estructura del entorno de programación

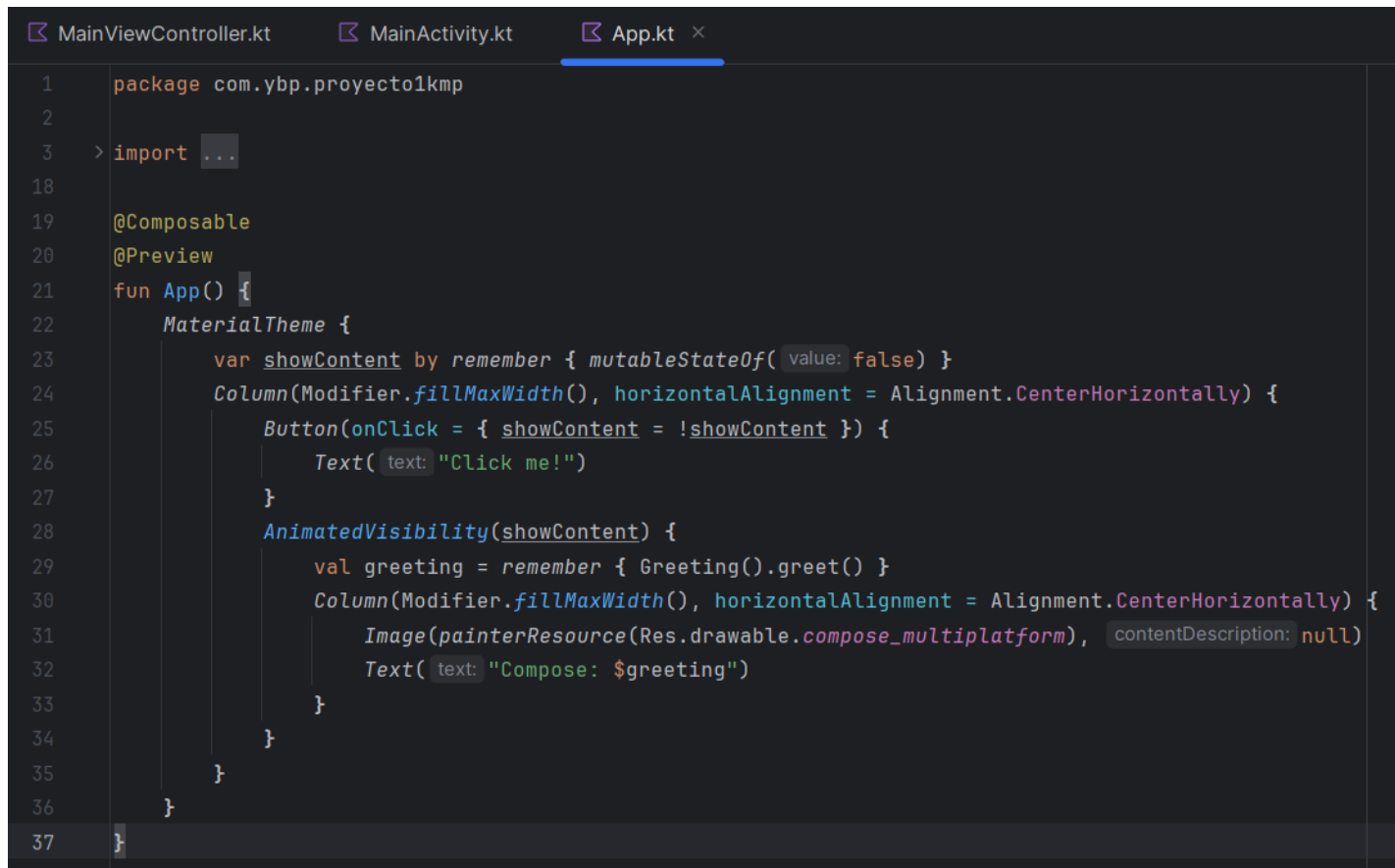
La programación de la aplicación se lleva acabo en la función **App()** , para lo cual abra la tomándola de alguna de las clases de las plataformas creadas. Ver figura 16.



```
1 package com.ybp.proyecto1kmp
2
3 > import ...
4
5
6
7
8
9 class MainActivity : ComponentActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12
13         setContent {
14             App()
15         }
16     }
17 }
18
19 @Preview
20 @Composable
21 fun AppAndroidPreview() {
22     App()
23 }
```

Fig. 16

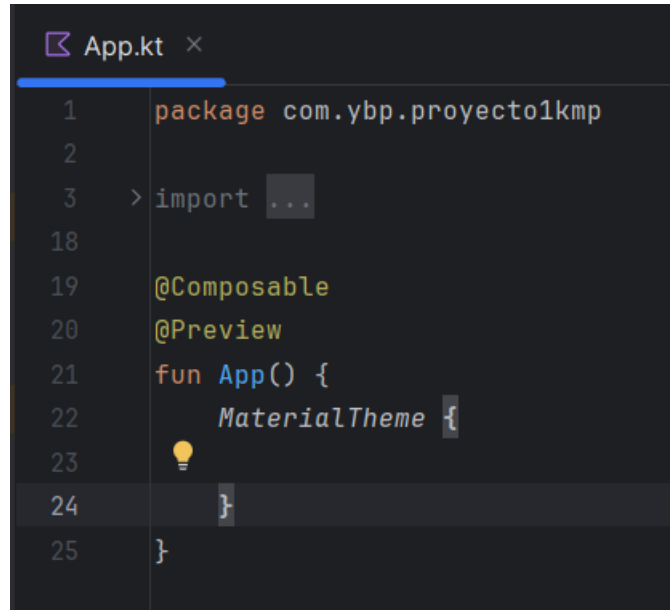
Para abrir la función App() que es donde se escribirá el código de programa ubique el puntero del mouse sobre la función App() y pulse a laves la tecla **Ctrl + Click** con el botón izquierdo del mouse, luego vera una ventana como la que describe en la figura 17.



```
1 package com.ybp.proyecto1kmp
2
3 > import ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19 @Composable
20 @Preview
21 fun App() {
22     MaterialTheme {
23         var showContent by remember { mutableStateOf( value: false) }
24         Column(Modifier.fillMaxWidth(), horizontalAlignment = Alignment.CenterHorizontally) {
25             Button(onClick = { showContent = !showContent }) {
26                 Text( text: "Click me!")
27             }
28             AnimatedVisibility(showContent) {
29                 val greeting = remember { Greeting().greet() }
30                 Column(Modifier.fillMaxWidth(), horizontalAlignment = Alignment.CenterHorizontally) {
31                     Image(painterResource(Res.drawable.compose_multiplatform), contentDescription: null)
32                     Text( text: "Compose: $greeting")
33                 }
34             }
35         }
36     }
37 }
```

Fig. 17

Posteriormente edite el código de la figura 17, como se muestra en la figura 18.



```
App.kt x
1 package com.ybp.proyecto1kmp
2
3 > import ...
18
19 @Composable
20 @Preview
21 fun App() {
22     MaterialTheme {
23         ⚡
24     }
25 }
```

Fig. 18

Empecemos a programar

Transcriba y ejecute los siguientes ejemplos:

Ejemplo1

```
fun App() {
    MaterialTheme {
        var horaLocal by remember { mutableStateOf("Hora No seleccionada") }
        Column {
            Text(horaLocal)
            Button(onClick = { horaLocal = "13:30" }) {
                Text("Mostrar hora Local")
            }
        }
    }
}
```

Ejemplo2

```
fun App() {  
    MaterialTheme {  
        Text("Caperucita roja")  
        Text("Chavo del Ocho")  
        Text("Chilindrina")  
    }  
}
```

Ejemplo3

```
fun App() {  
    MaterialTheme {  
        Column {  
            Text("Caperucita roja")  
            Text("Chavo del Ocho")  
            Text("Chilindrina")  
        }  
    }  
}
```

Ejemplo4

```
fun App() {  
    MaterialTheme {  
        Row {  
            Text("Caperucita roja")  
            Text("Chavo del Ocho")  
            Text("Chilindrina")  
        }  
    }  
}
```

Actividad

1. Crear una aplicación en Kotlin Multiplataforma, que permita realizar calcular el resultado de 8 y 10 numero.
2. Cree una pantalla que contenga la siguiente información:

Universidad de san Buenaventura, Bogotá

Facultad de Ingeniería de Sistemas

Asignatura: Desarrollo de Aplicaciones Móviles

Tema: Configuración Entorno Kotlin MultiPlataforma

Año: 2025-1