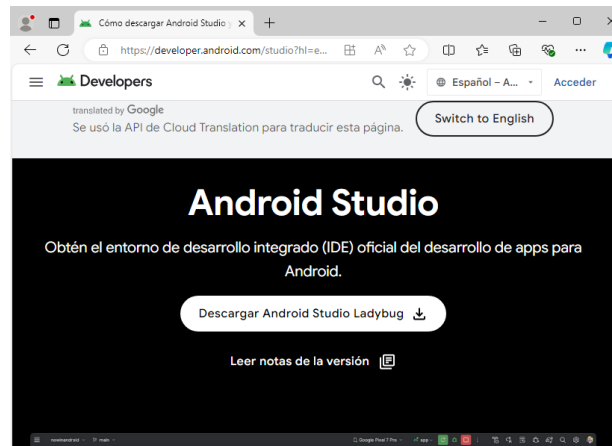




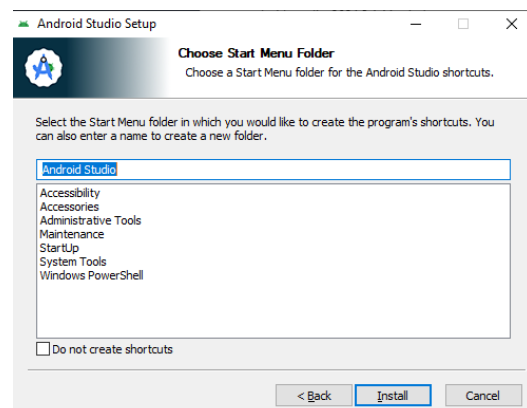
# **Presentación del Tema 1**

## Instalación de Android Studio

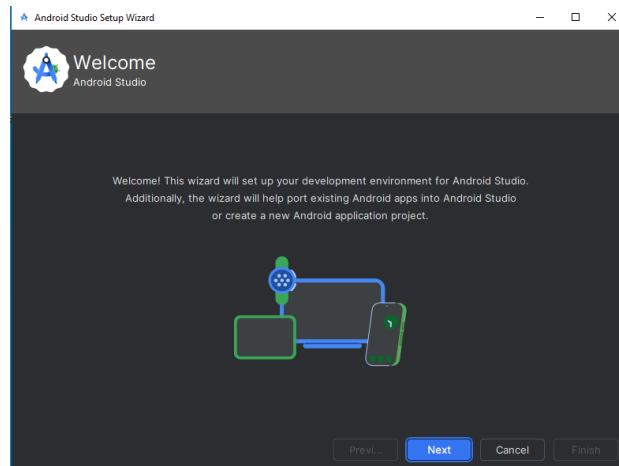
El primer paso será dirigirnos a `developer.android.com` y descargar el Android Studio.



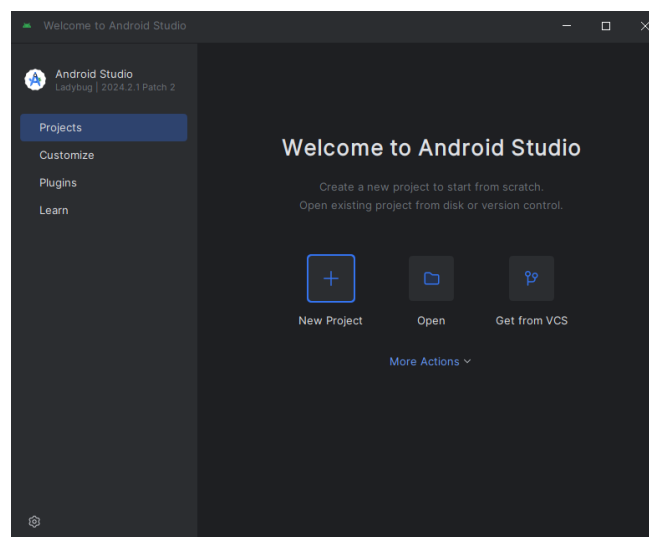
Continuamos una vez descargado con la instalación navegando por las siguientes ventanas hasta darle a “Install”:



Una vez instalado, nos mostrará esta ventana.



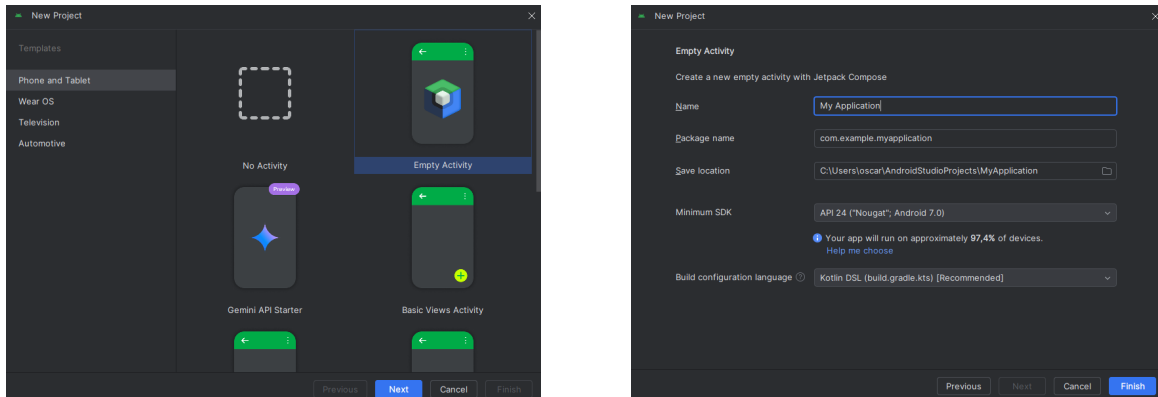
Aceptamos los términos y le damos “Next” hasta el final y dejamos que siga instalando paquetes.



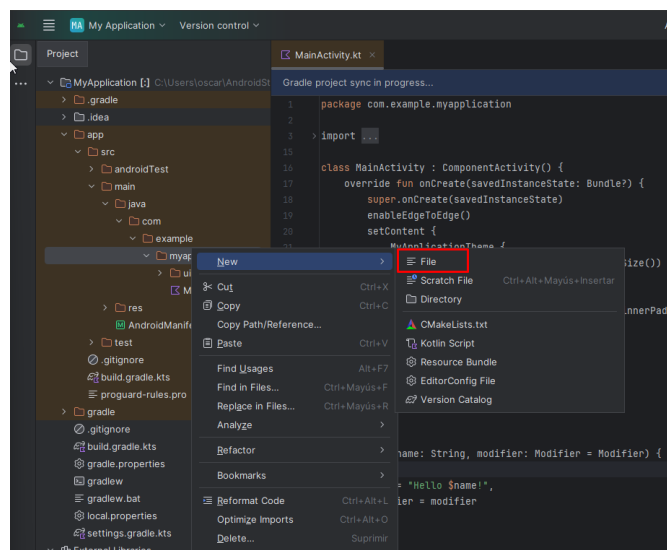
Y ya lo tendremos instalado, ahora podemos crear un proyecto.

## Uso del Entorno de Trabajo

Creamos un nuevo proyecto y seleccionamos “Empty Activity” y lo creamos.

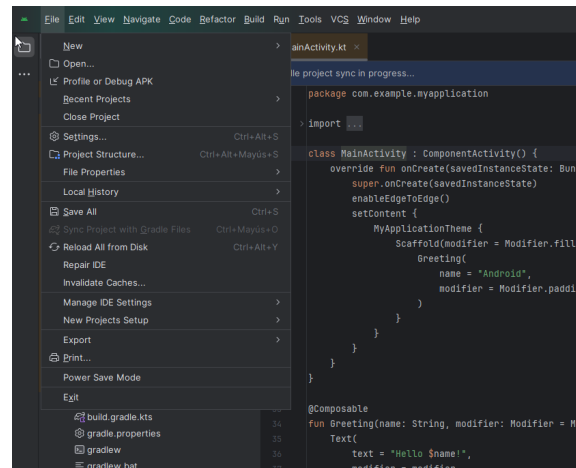


Para crear un nuevo archivo, hacemos click derecho en la carpeta en la que lo queramos crear, le damos a “New” y “New file”.



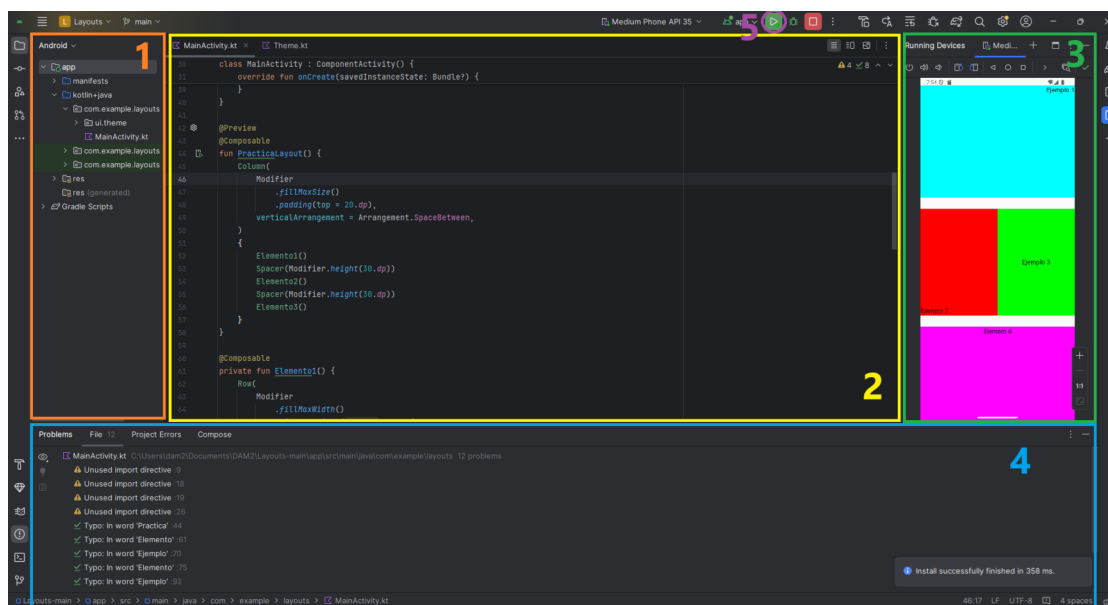
Para ver la barra de herramientas, le damos arriba a la izquierda al icono con cuatro líneas, y se nos abre la barra de herramientas donde podremos navegar a pestañas como “File”, “Edit”, “Navigate”, “View”, “Code”, etc.

Todos estos apartados nos serán útiles a la hora de desarrollar nuestro proyecto.



En la siguiente imagen, podemos ver la ventana principal de desarrollo la cual muestra lo siguiente:

1. El árbol de carpetas del proyecto: Permite navegar entre los distintos archivos.
2. El editor de texto: Permite visualizar y editar el código del proyecto.
3. El área de emulador: Aparece un móvil que muestra la App cuando la ejecutamos.
4. El área de consola y debug: Muestra la compilación y posibles errores de código.
5. Botón de ejecución: Comenzará la ejecución de nuestro código en el emulador.



## Ejemplo de Ejercicio

Este es un pequeño proyecto, el cual muestra el pantalla unas cajas de colores y unos textos. Este proyecto nos ayuda a entender como funciona la disposición en pantalla y como distribuir el contenido. Adjunto el código (sin imports que son muchas líneas):

```
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            LayoutsTheme {
                PracticaLayout()
            }
        }
    }
}

@Preview
@Composable
fun PracticaLayout() {
    Column(
        Modifier
            .fillMaxSize()
            .padding(top = 20.dp),
        verticalArrangement = Arrangement.SpaceBetween,
    ) {
        Elemento1()
        Spacer(Modifier.height(30.dp))
        Elemento2()
        Spacer(Modifier.height(30.dp))
        Elemento3()
    }
}

@Composable
private fun Elemento1() {
    Row(
        Modifier
            .fillMaxWidth()
            .fillMaxHeight(.33333f)
            .background(Color.Cyan),
        horizontalArrangement = Arrangement.End
    ) {
        Text("Ejemplo 1")
    }
}

@Composable
private fun Elemento2() {
    Row(
        Modifier
            .fillMaxWidth()
            .fillMaxHeight(.5f),
        verticalAlignment = Alignment.CenterVertically,
    ) {
        Column(
            Modifier
                .weight(1f)
                .fillMaxWidth()
                .fillMaxHeight()
                .background(Color.Red),
            verticalArrangement = Arrangement.Bottom
        )
    }
}
```

```

    {
        Text("Ejemplo 2")
    }
    Column(
        Modifier
            .weight(1f)
            .fillMaxWidth()
            .fillMaxHeight()
            .background(Color.Green),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    )
    {
        Text("Ejemplo 3")
    }
}

@Composable
private fun Elemento3() {
    Row(
        Modifier.Companion
            .fillMaxWidth()
            .fillMaxHeight()
            .background(Color.Magenta),
        horizontalArrangement = Arrangement.Center
    )
    {
        Text("Ejemplo 4")
    }
}

```

Vemos que el programa principal “MainActivity” ejecuta la función “PracticalLayout()”, la cual llama a Elemento 1, 2 y 3. Cada uno de estos elementos es una fila del resultado, siendo el 1 la fila celeste, el 2 la fila roja y verde y el 3 la fila rosa.

Este es el resultado:



## Error Común

Un error común sería que en “PracticaLayout()” no incluyéramos la línea “.padding(top = 20.dp)”, la cual haría que no se dejara un espacio en la parte superior, como podemos ver en este resultado:

