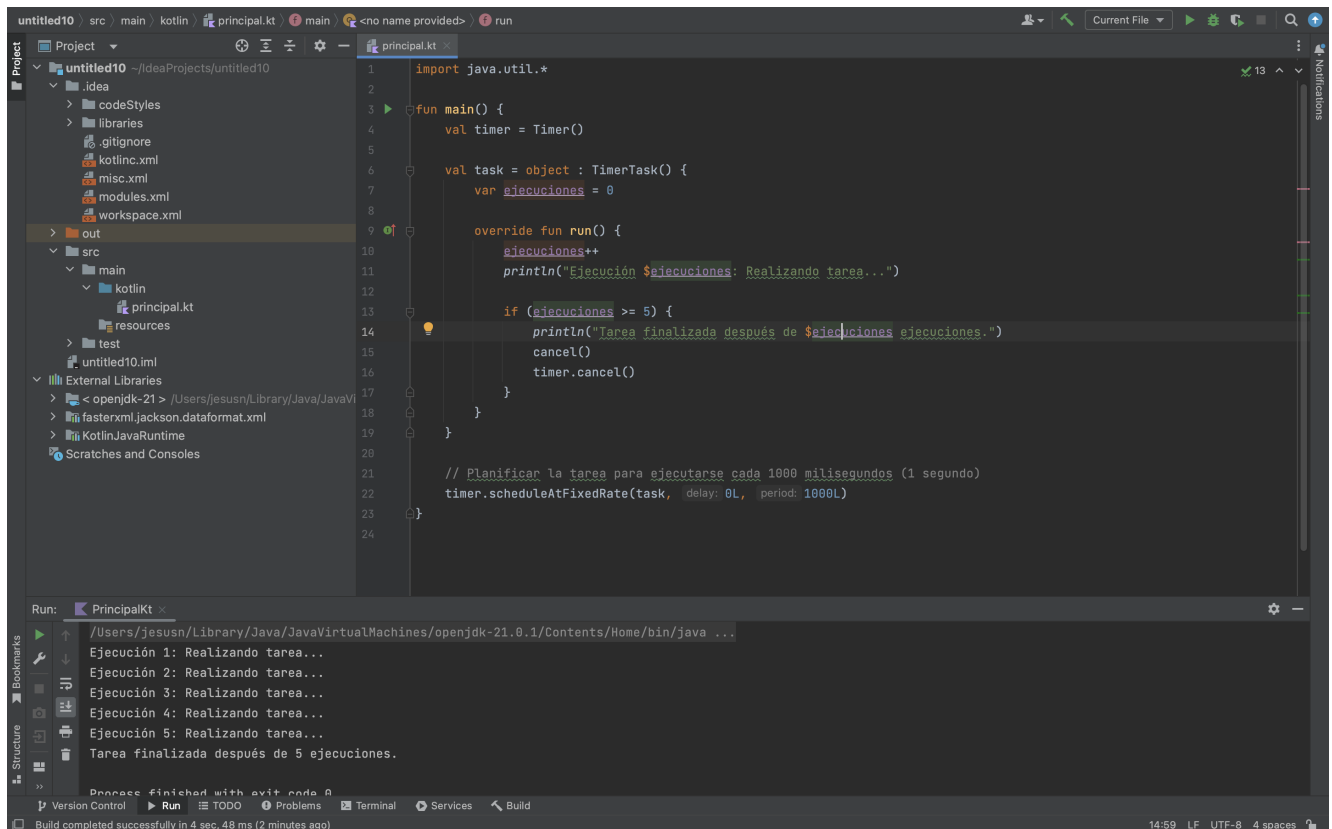


Ejemplo de uso de TimerTask en Kotlin

Envuelve la acción especificada en un TimerTask.

[crayon-6a9d68b4890ad671447967/]



The screenshot displays an IDE window with a Kotlin file named `principal.kt`. The code implements a `main` function that creates a `Timer` and a `TimerTask`. The `run` method of the task increments a counter and prints a message. When the counter reaches 5, it prints a completion message and cancels the timer. The task is scheduled to run every 1000 milliseconds. The `Run` console at the bottom shows the output of the program, confirming that the task executed 5 times before finishing.

```
1 import java.util.*
2
3 fun main() {
4     val timer = Timer()
5
6     val task = object : TimerTask() {
7         var ejecuciones = 0
8
9         override fun run() {
10             ejecuciones++
11             println("Ejecución $ejecuciones: Realizando tarea...")
12
13             if (ejecuciones >= 5) {
14                 println("Tarea finalizada después de $ejecuciones ejecuciones.")
15                 cancel()
16                 timer.cancel()
17             }
18         }
19     }
20
21     // Planificar la tarea para ejecutarse cada 1000 milisegundos (1 segundo)
22     timer.scheduleAtFixedRate(task, delay: 0L, period: 1000L)
23 }
24
```

Run: PrincipalKt

```
/Users/jesusn/Library/Java/JavaVirtualMachines/openjdk-21.0.1/Contents/Home/bin/java ...
Ejecución 1: Realizando tarea...
Ejecución 2: Realizando tarea...
Ejecución 3: Realizando tarea...
Ejecución 4: Realizando tarea...
Ejecución 5: Realizando tarea...
Tarea finalizada después de 5 ejecuciones.
Process finished with exit code 0
```