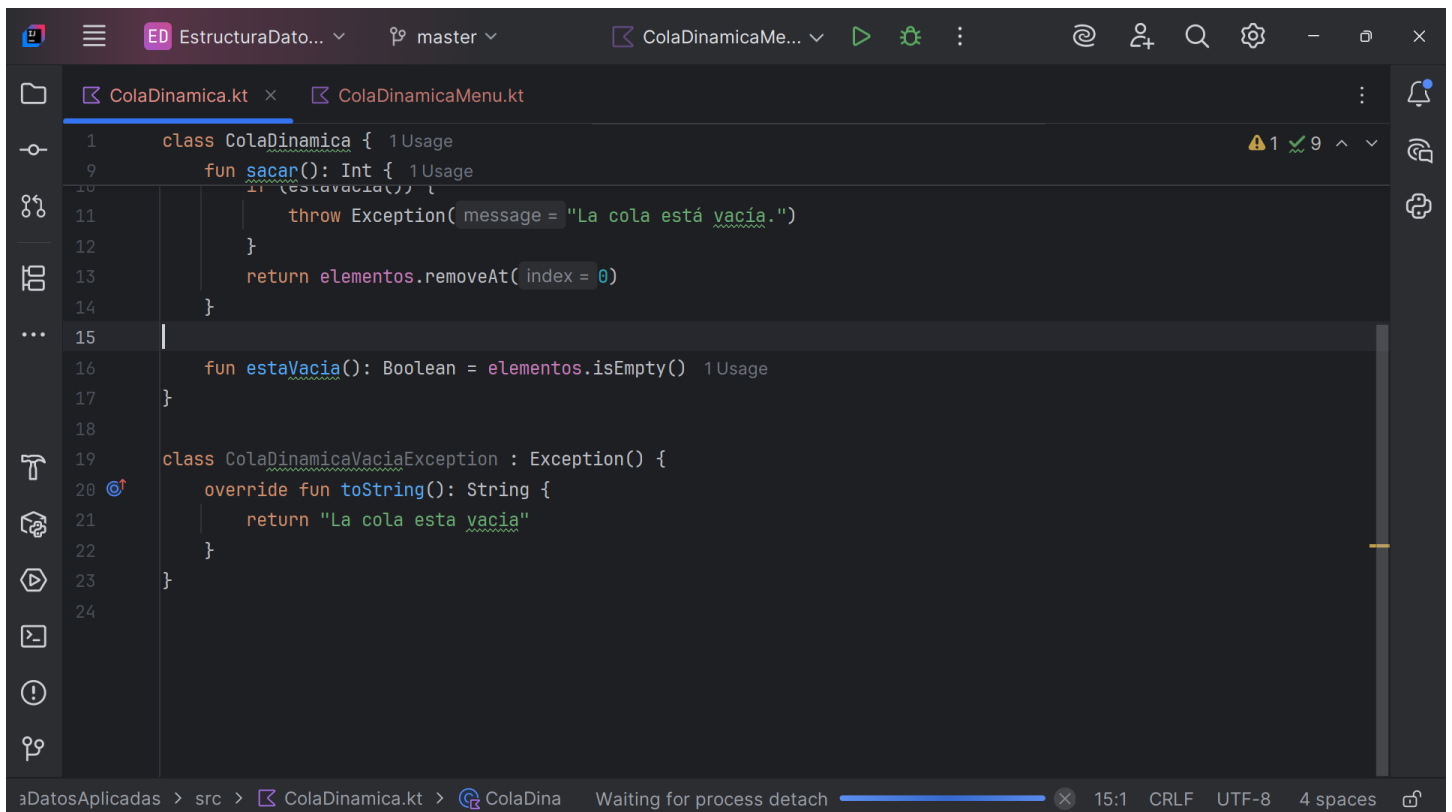


```
1 class ColaDinamica { 1 Usage
2     private val elementos = ArrayList<Int>() 3 Usages
3
4     fun meter(num: Int) { 1 Usage
5         elementos.add(num)
6         println("Elemento $num agregado a la cola.")
7     }
8
9     fun sacar(): Int { 1 Usage
10        if (estaVacia()) {
11            throw Exception(message = "La cola está vacía.")
12        }
13        return elementos.removeAt(index = 0)
14    }
15
16    fun estaVacia(): Boolean = elementos.isEmpty() 1 Usage
17 }
18
19 class ColaDinamicaVaciaException : Exception() {
20     override fun toString(): String {
21         return "La cola esta vacia"
22     }
23 }
```

src > ColaDinamica.kt > ColaDinamica Waiting for process detach 1:1 CRLF UTF-8 4 spaces



```
9     fun sacar(): Int { 1 Usage
10        if (estaVacia()) {
11            throw Exception(message = "La cola está vacía.")
12        }
13        return elementos.removeAt(index = 0)
14    }
15
16    fun estaVacia(): Boolean = elementos.isEmpty() 1 Usage
17 }
18
19 class ColaDinamicaVaciaException : Exception() {
20     override fun toString(): String {
21         return "La cola esta vacia"
22     }
23 }
```

src > ColaDinamica.kt > ColaDinamica Waiting for process detach 15:1 CRLF UTF-8 4 spaces

```
import kotlin.text.toIntOrNull

fun main() {
    val cola = ColaDinamica()
    var opcion: String?

    do {
        println("OPERACIONES EN LA COLA")
        println("1. METER")
        println("2. SACAR")
        println("3. SALIR")
        println("ELIGE UNA OPCIÓN: ")
        opcion = readLine()

        when (opcion) {
            "1" -> {
                print("Introduce un valor: ")
                val entrada = readLine()
                val numero = entrada?.toIntOrNull()
                if (numero != null) {
                    cola.meter( num = numero)
                }
            }
        }
    } while (opcion != "3")
}
```

```
fun main() {
    val numero = entrada?.toIntOrNull()
    if (numero != null) {
        cola.meter( num = numero)
    } else {
        println("Entrada no valida")
    }

    "2" -> {
        try {
            val elemento = cola.sacar()
            println("Elemento que sale: $elemento")
        } catch (e: Exception) {
            println(e.message)
        }
    }

    "3" -> println("Saliendo del programa...")
    else -> println("Opcion invalida intenta de nuevo")
} while (opcion != "3")
}
```

```
C:\Users\luffy\.jdk\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition\lib\idea_rt.jar=1273.1273:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition\bin" -Dfile.encoding=UTF-8
OPERACIONES EN LA COLA
1. METER
2. SACAR
3. SALIR
ELIGE UNA OPCIÓN:
1
Introduce un valor: 10
Elemento 10 agregado a la cola.
OPERACIONES EN LA COLA
1. METER
2. SACAR
3. SALIR
ELIGE UNA OPCIÓN:
1
Introduce un valor: 20
Elemento 20 agregado a la cola.
OPERACIONES EN LA COLA
1. METER
```

atosAplicadas > src > ColaDinamicaMenu.kt > main Waiting for process detach 9:28 CRLF UTF-8 4 spaces

```
1. METER
2. SACAR
3. SALIR
ELIGE UNA OPCIÓN:
1
Introduce un valor: 30
Elemento 30 agregado a la cola.
OPERACIONES EN LA COLA
1. METER
2. SACAR
3. SALIR
ELIGE UNA OPCIÓN:
2
Elemento que sale: 10
OPERACIONES EN LA COLA
1. METER
2. SACAR
3. SALIR
```

atosAplicadas > src > ColaDinamicaMenu.kt > main Waiting for process detach 9:28 CRLF UTF-8 4 spaces

