

Enviar un mensaje TCP entre un cliente en Swift para un Apple Watch y un servidor web en Kotlin que se ejecuta en IntelliJ IDEA y que lo reproduce con la voz del sistema

Servidor en Kotlin

Importar librerías

[crayon-6a9d4a2048a00491109436/]

Código del servidor que escucha las peticiones y las reproduce con la voz del sistema

[crayon-6a9d4a2048a05366035829/]

Cliente en Swift para el Apple Watch que envía mensajes

[crayon-6a9d4a2048a08547060093/]

The screenshot shows an IDE with a project named 'untitled20'. The main file, 'servidro.kt', contains the following Kotlin code:

```
1 import io.ktor.application.*
2 import io.ktor.http.*
3 import io.ktor.request.*
4 import io.ktor.response.*
5 import io.ktor.routing.*
6 import io.ktor.server.engine.*
7 import io.ktor.server.netty.*
8 import java.io.File
9
10 fun main() {
11     val server = embeddedServer(Netty, port = 6001) { this: Application
12         routing { this: Routing
13             post(path = "/send-message") { this: PipelineContext<Unit, ApplicationCall>
14                 val receivedMessage = call.receiveText()
15                 println("Mensaje recibido: $receivedMessage")
16
17                 val cmd = arrayOf("say", receivedMessage) // Comando 'say' para macOS, cámbialo según tu sistema operativo
18                 Runtime.getRuntime().exec(cmd)
19
20                 call.respond(HttpStatusCode.OK)
21             }
22         }
23     }
24     server.start(wait = true)
25 }
```

The bottom panel shows the 'Run' output with the following messages:

```
Run: ServidroKt
Starting info ktor.application - Application started in 0.004 seconds.
Mensaje recibido: hola amigos
Mensaje recibido: hola amigos
Mensaje recibido: hola amigos
Mensaje recibido: hola amigos
Mensaje recibido: hola amigos
```

