Entendido. Si estás usando un módulo WiFi en tu Arduino para la comunicación con el servidor Node.js a través de WebSocket, no necesitas la comunicación serial. Aquí está la adaptación del código para enviar mensajes al Arduino a través de la conexión WebSocket:

**En el archivo server.js:**

const http = require('http');

const express = require('express');

const socketIo = require('socket.io');

const app = express();

const server = http.createServer(app);

const io = socketIo(server);

io.on('connection', (socket) => {

console.log('Nuevo cliente conectado');

// Manejar evento para activar el riego por un número específico de segundos

socket.on('activarRiegoSegundos', (segundos) => {

console.log(`Activando riego durante ${segundos} segundos`);

// Enviar mensaje al Arduino para activar el riego

socket.emit('activarRiego');

// Programar el apagado del riego después de ciertos segundos

setTimeout(() => {

// Lógica para apagar el riego después de ciertos segundos

console.log('Apagando riego');

// Enviar mensaje al Arduino para apagar el riego

// Este mensaje depende de cómo lo manejas en el Arduino

}, segundos \* 1000); // Convertir segundos a milisegundos

});

// Puedes definir más eventos según sea necesario

socket.on('disconnect', () => {

console.log('Cliente desconectado');

});

});

const PORT = process.env.PORT || 3000;

server.listen(PORT, () => {

console.log(`Servidor WebSocket iniciado en el puerto ${PORT}`);

});

**En el código del Arduino:**

#include <ESP8266WiFi.h>

#include <WebSocketsClient.h>

const char\* ssid = "TU\_SSID";

const char\* password = "TU\_CLAVE\_WIFI";

const char\* webSocketServerAddress = "IP\_DEL\_SERVIDOR\_WS";

WebSocketsClient webSocket;

void setup() {

Serial.begin(115200);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.println("Conectando a la red WiFi...");

}

Serial.println("Conexión WiFi establecida.");

webSocket.begin(webSocketServerAddress);

webSocket.onEvent(webSocketEvent);

}

void loop() {

webSocket.loop();

}

void webSocketEvent(WStype\_t type, uint8\_t \* payload, size\_t length) {

switch(type) {

case WStype\_DISCONNECTED:

Serial.println("Desconectado del servidor WebSocket");

break;

case WStype\_CONNECTED:

Serial.println("Conectado al servidor WebSocket");

break;

case WStype\_TEXT:

// Manejar mensajes de texto recibidos del servidor

if (strcmp((char\*)payload, "activarRiego") == 0) {

Serial.println("Activando riego");

// Ejecutar la lógica para activar el riego en el Arduino

activarRiego();

}

break;

case WStype\_BIN:

// Manejar mensajes binarios si es necesario

break;

}

}

void activarRiego() {

// Lógica para activar la bomba de agua durante un número específico de segundos

digitalWrite(PIN\_DE\_LA\_BOMBA, HIGH); // Enciende la bomba de agua

delay(5000); // Activa la bomba durante 5 segundos (ajusta el tiempo según sea necesario)

digitalWrite(PIN\_DE\_LA\_BOMBA, LOW); // Apaga la bomba de agua

}