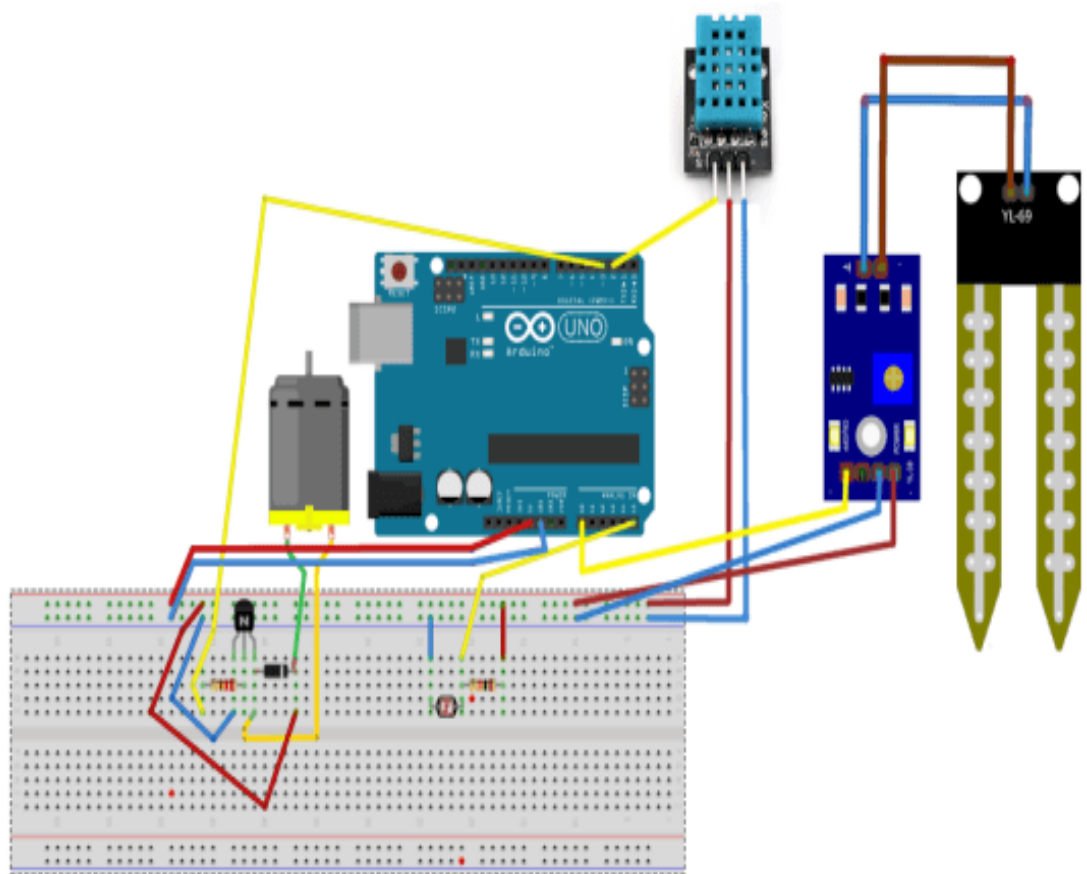




**CIFP DON BOSCO**  
**DPTO. ELECTRÓNICA**  
**Sistema de Riego Automático**



**Lander Hernández**  
**Amadeo Nguema**  
**Miguel Chilquillo**

# Índice

- Introducción
- Análisis de materiales
- Programa Arduino
- Esquema eléctrico
- Problemas
- Mejoras
- Conclusión
- Bibliografía

# Introducción

Estamos acabando nuestro último año del grado y nos plantearon como trabajo de fin de curso un proyecto que sería un sistema de riego automático con arduino y sensores entre otras cosas.

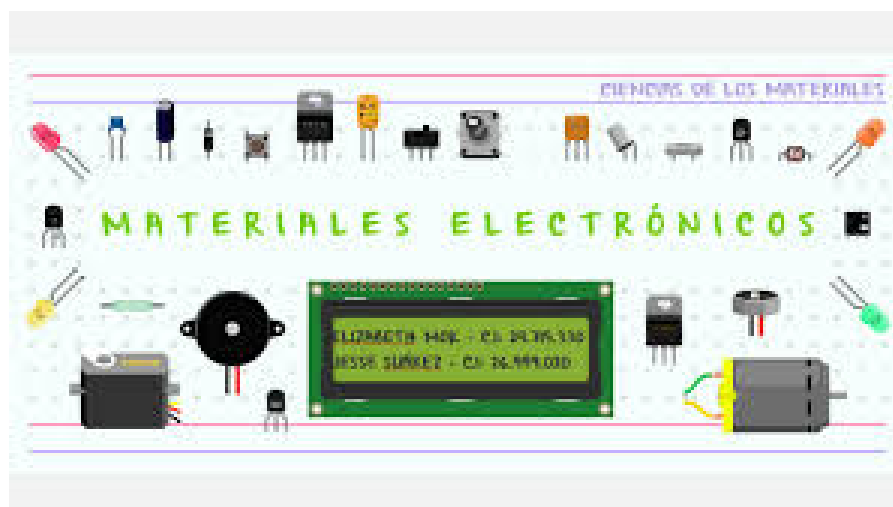
Para este proyecto vamos a usar 4 sensores de humedad, 1 sensor de temperatura, 1 sensor de luz (fotoresistor) y 1 motobomba que bombeará agua y con ella regaremos los 4 recipientes que tenemos para simular las plantas, cada recipiente llevará su propia válvula para poder regar cada una individualmente en el caso que sea necesario. Para ello tenemos la ayuda de un grado de informática de Zubiri, los cuales nos ayudarán con la página web y con el servidor para poder reflejar ahí los datos que vayamos recibiendo de los sensores, los datos que recibiremos serán los datos de la humedad de la tierra, la temperatura y la luz.

Con ayuda de MQTT, NodeRed y NodeMCU hemos hecho un servidor donde llegarán todos los datos que recibamos y desde la página web que nos han diseñado podremos hacer cambios en los datos y podremos regar las plantas cuando queramos indistintamente sea una planta, dos o las cuatro.

*Por ejemplo: Tenemos definido que cuando el sensor de temperatura note que baja más de 20° riegue (también se puede hacer con las variables de la humedad), pues desde el teléfono podremos cambiar esas variables y poner que cuando esté a 22° riegue. También otro ejemplo sería que una de las plantas esté “seca” y manualmente poder regar la planta sin tener que esperar a que pase el periodo de tiempo para el siguiente riego y eso lo haríamos con la ayuda de las válvulas que nos haría la función de un relé (deja pasar el agua o no).*

# Análisis de Materiales

- Arduino NANO
- Protoboard
- Sensor de Humedad
- Sensor de Temperatura
- Motobomba
- Fotorresistor
- Tubos
- Raspberry pi 3 B+
- MQTT
- Node-RED
- NodeMCU
- Pantalla Raspberry pi táctil



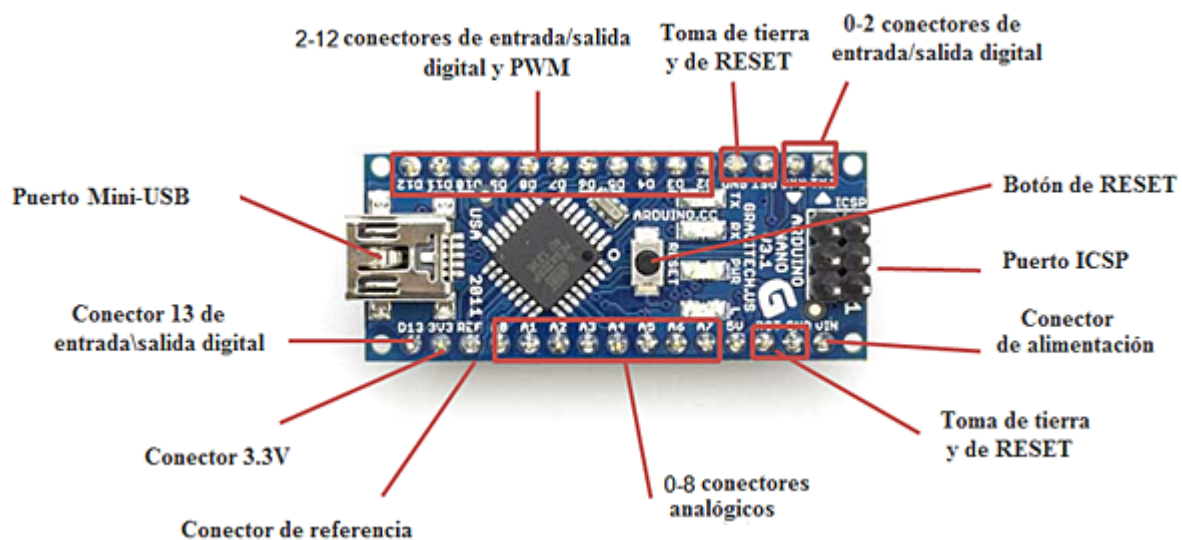
Para que un componente o sensor funcione, requiere que lo programes en el código de Arduino. Vamos a ir viendo cómo plantear las especificaciones según el componente y los requerimientos

Idea	Requerimiento
Microcontrolador: conexión a Internet	Al necesitar conexión a Internet necesitamos conexión con WiFi pero, ¿es necesario desde el primer momento?
Temperatura	Dependerá de donde vivas. No es lo mismo medir la temperatura en Sevilla que en Estocolmo . El rango que debe medir es totalmente diferente.
Humedad	Puedes usar un sensor específico para medir la humedad o uno combinado para medir temperatura y humedad a la vez
Presión atmosférica	Igual que en el caso anterior, ¿existe algún sensor que mida temperatura, humedad y presión?
Luminosidad	Con una resistencia LDR te puede servir, dependerá de dónde vivas
Batería	¿Qué consumo necesita el sistema? ¿Soporta la alimentación con baterías? ¿Voltaje de operación?
Pantalla información	¿Qué información hay que mostrar? Temperatura, humedad, presión, decibelios, contaminación, nivel de batería, velocidad del viento, etc....

Plataforma en la nube	¿Realmente es útil para ti almacenar la información? ¿toda la información?
Aplicación web	Debe ser responsivo para que se vea en todos los dispositivos.

## Arduino NANO

El Arduino Nano es una pequeña y completa placa basada en el ATmega328. Es muy compacto y puede ser utilizado fácilmente en un protoboard. Tiene más o menos la misma funcionalidad que el Arduino Uno, pero con una presentación diferente. La placa es compatible 100%.



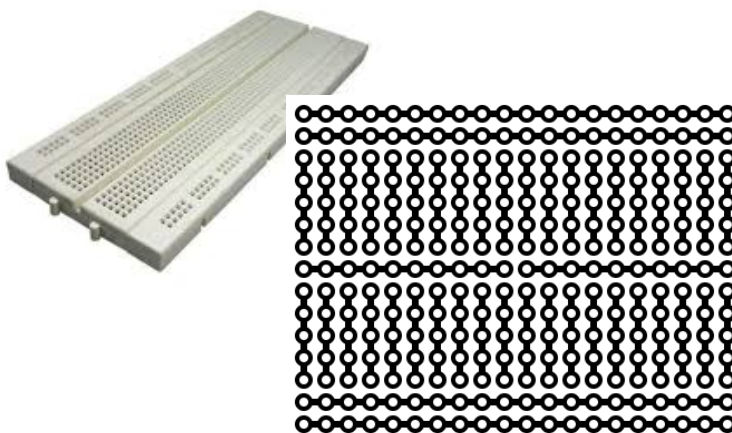
### Especificaciones técnicas de Arduino Nano

- Microcontrolador Arduino ATmega328
- Arquitectura, AVR
- Voltaje de operación, 5 V
- Memoria flash, 32 KB de los cuales 2 KB utilizados por bootloader
- SRAM 2 KB

- Velocidad del reloj 16 MHz
- Pines de E/S analógicas, 8
- EEPROM, 1 KB
- Corriente continua por pin entrada salida, 40 mA (Pines de E/S)
- Voltaje de entrada, 7-12V
- Pines de E/S digitales, 22
- Salida PWM, 6
- Consumo de energía, 19 mA
- Tamaño de la placa de circuito impreso, 18 x 45 mm
- Peso, 7 g

## ProtoBoard

El protoboard es una placa que posee unos orificios conectados eléctricamente entre sí siguiendo un patrón horizontal o vertical. Es empleada para realizar pruebas de circuitos electrónicos, insertando en ella componentes electrónicos y cables como puente. Es el boceto de un circuito electrónico donde se realizan las pruebas de funcionamiento necesarias antes de trasladarlo sobre un circuito impreso. Esta placa puede llamarse de varias formas, las más comunes son «*protoboard*», «*breadboard*», «*placa protoboard*» o incluso «*placa de pruebas*».

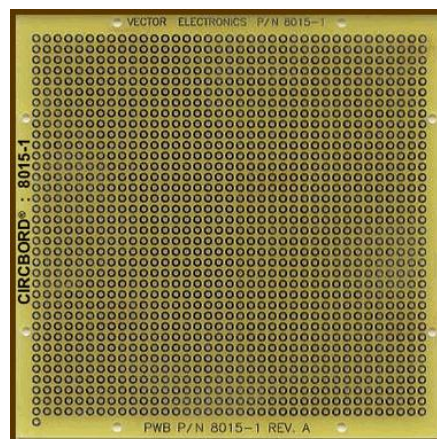


-Esta suele tener un patrón típico de disposición de las láminas de material conductor en una placa de pruebas.

## Otros Tipos de placas:

### Placa de perforación

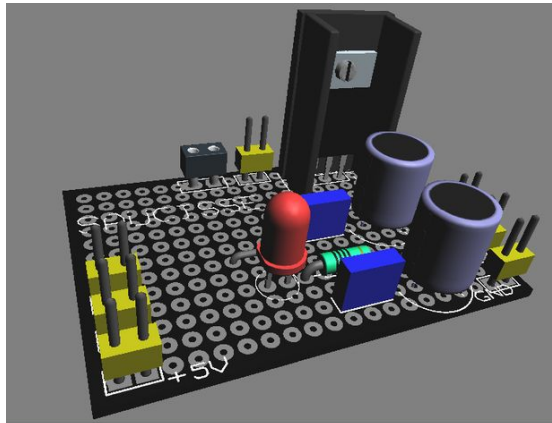
Pega los cables de los componentes a través de los orificios y suelda los cables de un componente a los cables de otros componentes. Tiene un patrón de cobre impreso en su superficie, al que puede soldar componentes. El tipo más simple es la almohadilla por orificio, que simplemente tiene un anillo de cobre alrededor de cada orificio. La ventaja de esto sobre el tablero de perforación simple es que cada componente está soldado al tablero en lugar de entre sí, por lo que es más fácil trabajar con él y el proyecto es más estable.



### Perfboard complicado

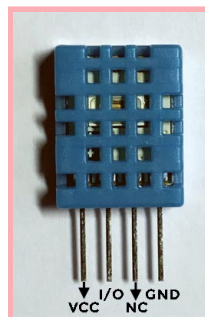
Tipos de tableros perfilados con varios patrones de cobre que conectan múltiples orificios. En la imagen de abajo, las marcas blancas en la parte superior del tablero le indican dónde está el cobre en la parte inferior. Este tipo en particular tiene un patrón complicado: hay dos tiras de bus en el centro de la placa, almohadillas de 2 orificios en los bordes de la placa y almohadillas de 3 orificios y almohadillas por orificios entre las dos.





## Sensor de Temperatura

El DHT11 es un sensor digital de temperatura y humedad de bajo costo y fácil uso. Integra un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos (no posee salida analógica). Una de las ventajas que nos ofrece el DHT11, además de medir la temperatura y la humedad, es que es digital. A diferencia de sensores como el LM35, este sensor utiliza un pin digital para enviarnos la información y por lo tanto, estaremos más protegidos frente al ruido.



Los pines de la versión sin PCB del DHT11 son:

- VCC: Alimentación
- I/O: Transmisión de datos
- NC: No conectado, pin al aire
- GND: Conexión a tierra

## Características técnicas del DHT11

Modelo DHT11	
Alimentación	De 3,5V a 5V
Consumo	2,4mA
Señal de salida	Digital

TEMPERATURA	
Rango	De 0°C a 50°C
Precisión	A 25°C $\pm$ 2°C
Resolución	1°C (8-bit)

HUMEDAD	
Rango	De 20% RH a 90%
Precisión	Entre 0°C y 50°C $\pm$ 5% RH
Resolución	1% RH

## ¿Cómo transmite los datos el DHT11?

El propio dispositivo hace la conversión entre analógico y digital. Por lo tanto, partimos de una señal analógica que luego es convertida en formato digital y se enviará al microcontrolador. La trama de datos es de 40 bits correspondiente a la información de humedad y temperatura del DHT11.

<u>0011 0101</u>	<u>0000 0000</u>	<u>0001 1000</u>	<u>0000 0000</u>	<u>0100 1001</u>
8 bits humedad	8 bits humedad	8 bits temperatura	8 bits temperatura	bits de paridad

Estos bits de paridad lo único que hacen es asegurarnos de que la información es correcta, sumando los 4 primero grupos de 8-bit. Esta suma debe ser igual a los bit de paridad. Si nos centramos en la imagen anterior y sumamos los bits, comprobamos que todo está correcto.

$$0011\ 0101 + 0000\ 0000 + 0001\ 1000 + 0000\ 0000 = 0100\ 1101$$

## Sensor YL-69

Este sensor tiene la capacidad de medir la humedad del suelo. Aplicando una pequeña tensión entre los terminales del módulo YL-69 hace pasar una corriente que depende básicamente de la resistencia que se genera en el suelo y ésta depende mucho de la humedad. Por lo tanto, al aumentar la humedad la corriente crece y al bajar la corriente disminuye.



## Características técnicas del YL-69

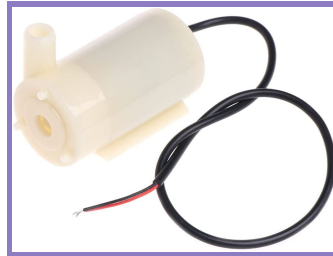
### Especificaciones

- Voltaje de entrada: 3.3 – 5 VCD
- Voltaje de salida: 0 ~ 4.2 V
- Corriente: 35 mA
- Dimensiones YL-38: 30 x 16 mm
- Dimensiones YL-69: 60 x 30 mm
- Peso: 7 g

### Terminales

- VCC: Tensión de alimentación
- GND: Tierra
- A0: Salida analógica que entrega una tensión proporcional a la humedad.
- D0: Salida digital; este módulo permite ajustar cuándo el nivel lógico en esta salida pasa de bajo a alto mediante el potenciómetro.

## Mini Bomba de Agua 3V



- Material: Plástico
  - Tipo: dc3 V-5 V/sumergible
  - Corriente: 100 – 200 mA
  - Máx. rendimiento: 100 L/h H. Max: 0,8 m
- Es ajustable y puede aumentar el oxígeno de su acuario.
  - Es pequeño, ahorro de espacio, se puede ajustar a cualquier lugar del depósito
  - Trabajo en conjunto con una bomba de aire, no echa de ruido.

## Fotorresistor

Una fotorresistencia es un componente eléctrico, el cual posee una resistencia capaz de variar su magnitud al estar en contacto con distintas magnitudes de intensidad lumínica.

Está conformado por una célula fotorreceptora y dos pastillas; cuando dicha célula no percibe cierto nivel de luz (es decir cuando baja del mínimo de lúmenes predefinido), entonces hace contacto y se encienden las luces.

### ¿Cómo funciona una fotorresistencia?

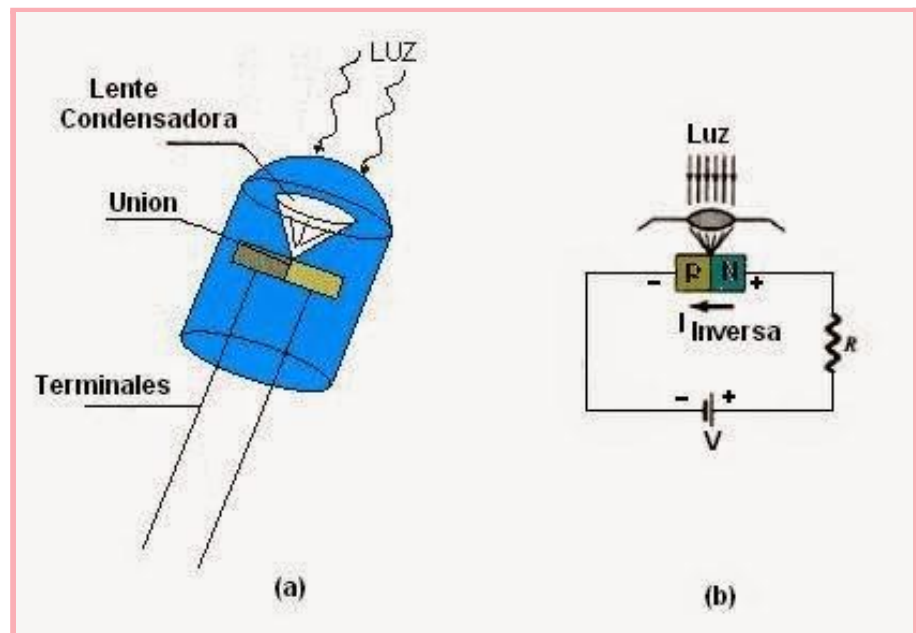
La base del funcionamiento de una fotorresistencia radica en su componente principal, el sulfuro de cadmio (CdS). Este componente químico es un semiconductor que tiene la capacidad de variar su resistencia según la cantidad de luz que en él incide.

Cuanto mayor intensidad es la luz que incide sobre el sulfuro de cadmio, más baja es la resistencia, es decir mayor facilidad de los electrones para moverse. Vale saber que la variación de la resistencia cuando hay cambios de luminosidad repentinos no sigue la misma velocidad, sino que tiene retardo.

## Tipos de fotoresistencia

Existen 2 tipos de fotorresistencias según la forma en que polarizan.

1. *Fotorresistencia o LDR lineales:* Son aquellas que polarizan a la inversa de la fuente donde se conecte. Son también llamadas fotodiodos.



2. *Fotorresistencia o LDR no lineales:* La polaridad de este fotorresistor no depende de la fuente donde se conecte.

## Características de las fotorresistencias

- Conformados por un semiconductor de alta resistencia.
- La resistencia varía entre  $1\text{M}\Omega$ , cuando hay mucha oscuridad y  $100\Omega$  con altas intensidades de luz

- Tiempo de respuesta de 1 décima de segundo cuando varía la intensidad lumínica.
- De muy bajo costo y tecnología sencilla, por lo tanto se puede encontrar en cámaras, medidores de luz, alarmas de seguridad, medidores de radio o los sistemas de alumbrado público; también en autos modernos.

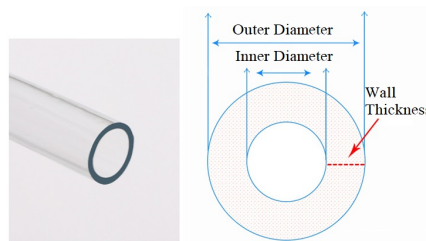
## ¿Cómo conectar una fotorresistencia?

Conectar una fotorresistencia es tan simple como conectar sus terminales al circuito, como si fuera una resistencia, la cual variará según la intensidad lumínica. En el siguiente video podemos ver paso a paso cómo hacer para conectar una fotorresistencia a un circuito.

Ventajas	Desventajas
Puede abarcar superficies grandes.	Se debe elegir dependiendo del tipo de luz que se necesita.
Fácil de conectar.	Se produce el efecto histéresis. Que hace dejar guardado en lo que sería en "la memoria" del metal ciertas condiciones que al cambiar la luz retrasa el funcionamiento.
Bajo costo.	No se puede usar en sitios donde la luz varía rápidamente.
Tiene una gran relación luz, oscuridad, electricidad.	La variación de la resistencia en función de la intensidad lumínica no es lineal.

## Tubos

Tubos de plástico por donde pasará el agua para regar los cuatro recipientes que usaremos en nuestro proyecto.



## RASPBERRY Pi 3 B+

La Raspberry Pi es una serie de **ordenadores de placa reducida**, ordenadores de placa única u ordenadores de placa simple (SBC).

No se indica expresamente si es hardware libre o con derechos de marca, en su web oficial explican que disponen de contratos de distribución y venta con dos empresas, pero al mismo tiempo cualquiera puede convertirse en revendedor o distribuidor de las tarjetas Raspberry Pi, por lo que da a entender que es un producto con propiedad registrada.

El software sí es de código abierto, siendo su sistema operativo oficial una versión adaptada de Debian, denominada Raspberry Pi OS, aunque permite usar otros sistemas operativos, incluido una versión de Windows 10. En todas sus versiones, incluye un procesador Broadcom, memoria RAM, GPU, puertos USB, HDMI, Ethernet (el primer modelo no lo tenía), 40 pines GPIO (desde la Raspberry Pi 2) y un conector para cámara.



## Pi 3 B+

Este nuevo micro-ordenador ha supuesto un rediseño absoluto de la placa, manteniendo el mismo tamaño y la misma posición de los elementos que en el modelo Pi 3, pero ha cambiado el procesador por otro más potente que funciona a 1.4 Ghz, y además elimina el cuello de botella de la conectividad incluyendo Bluetooth 4.2, BLE, Wi-Fi a doble banda 2.4 Ghz y 5 Ghz y, además, la tarjeta de red, Gigabit Ethernet, ya no está limitada a los 100 Mbps, sino que es capaz de alcanzar los 300 Mbps al funcionar sobre USB 2.0.



## TIPOS DE RASPBERRY

Especificaciones	Raspberry Pi 1 modelo A (descontinuada)	Raspberry Pi 1 modelo B (descontinuada)	Raspberry Pi 1 modelo B+	Raspberry Pi 2 modelo B	Raspberry Pi 3 modelo B	Raspberry Pi 3 modelo B+
SoC (CPU, GPU, DSP, RAM y puertos USB)	Broadcom BCM2835			Broadcom BCM2836	Broadcom BCM2837	
CPU	ARM 1176JZF-S a 700 MHz (familia ARM11)			900 MHz quad-core ARM Cortex A7	1.2GHz 64-bit quad-core ARMv8	1.4GHz 64-bit quad-core ARMv8

Juego de instrucciones	RISC de 32 bits			RISC de 64 bits	
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, MPEG-2 y VC-1 (con licencia), 1080p30 H.264/MPEG-4 AVC				
Memoria	256 MB (compartidos con la GPU)	512 MiB (compartidos con la GPU)		1 GB (compartidos con la GPU)	
Puertos USB 2.0	1	2 (vía hub USB integrado)	4		
Puertos USB 3.0	Ninguno				
Entradas de vídeo	Conector MIPI CSI que permite instalar un módulo de cámara desarrollado por la Raspberry Pi Foundation				
Salidas de vídeo	Conector RCA (PAL y NTSC), HDMI (rev1.3 y 1.4), Interfaz DSI para panel LCD				
Salidas de audio	Jack de 3.5 mm, HDMI				
Almacenamiento	SD / MMC / ranura para SDIO		MicroSD		
Conectividad de red	Ninguna	10/100 Ethernet (RJ-45) vía hub USB		Puerto RJ-45 (ethernet) de 10/100Mbps vía hub USB Wi-Fi 802.11bgn Bluetooth 4.1	Puerto RJ-45 (Ethernet) de 10/100/1000 Mbps vía hub USB limitado a 300Mbit/s Wi-Fi 802.11ac de doble banda

				Bluetooth 4.2 BLE
Periféricos de bajo nivel	8 x GPIO, SPI, I <sup>2</sup> C, UART			17 x GPIO y un bus HAT ID
Consumo energético	500 mA (2.5 W)	700 mA (3.5 W)	600 mA (3.0 W)	800 mA (4.0 W)
Fuente de alimentación	5 V vía Micro USB o puerto GPIO			
Dimensiones	85mm x 53mm			
Sistemas operativos soportados	GNU/Linux: Raspbian, Fedora (Pidora), Arch Linux (Arch Linux ARM), Slackware Linux, SUSE Linux Enterprise Server for ARM, RISC OS			

## CARACTERÍSTICAS:

- CPU + GPU: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- RAM: 1GB LPDDR2 SDRAM
- Wi-Fi + Bluetooth: 2.4GHz y 5GHz IEEE 802.11.b/g/n/ac, Bluetooth 4.2, BLE
- Ethernet: Gigabit Ethernet sobre USB 2.0 (300 Mbps)
- GPIO de 40 pines
- HDMI
- 4 puertos USB 2.0
- Puerto CSI para conectar una cámara.
- Puerto DSI para conectar una pantalla táctil
- Salida de audio estéreo y vídeo compuesto
- Micro-SD
- Power-over-Ethernet (PoE)

## MQTT

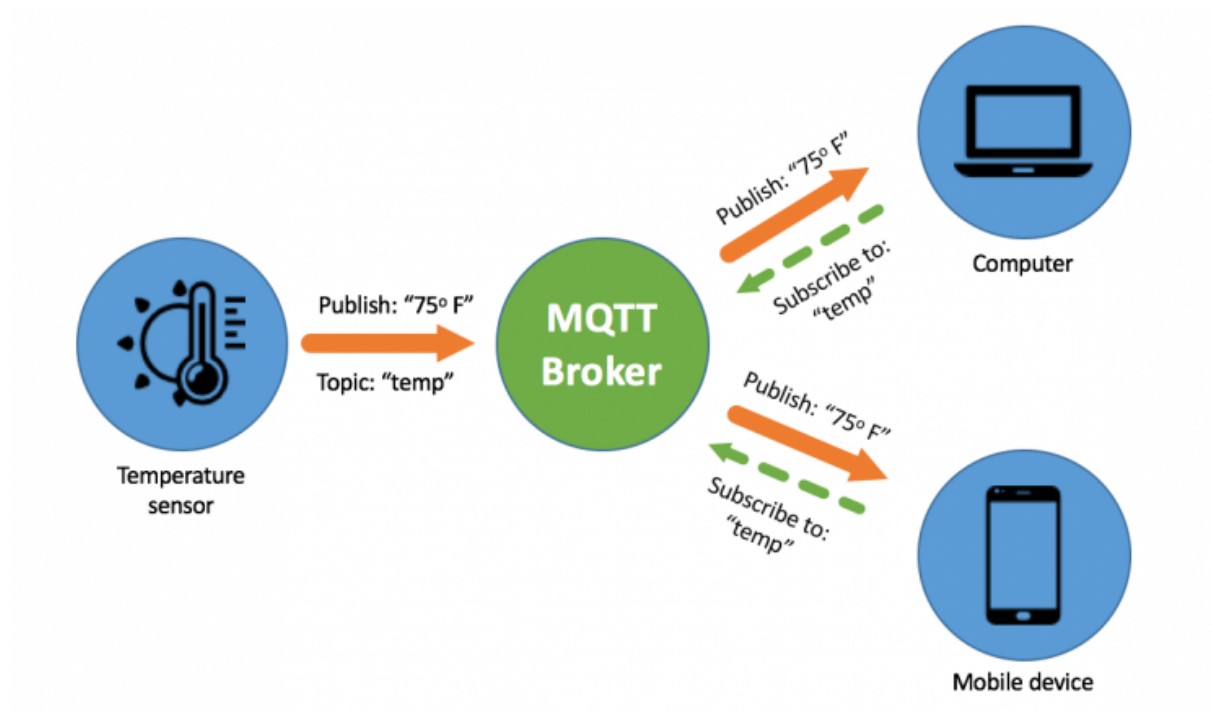
El estándar de mensajería de IOT. Es un estándar abierto OASIS e ISO ligero. El protocolo MQTT define dos tipos de entidades de red: un intermediario de mensajes y varios clientes. Un intermediario MQTT es un servidor que recibe todos los mensajes de los clientes y luego enruta los mensajes a los clientes de destino adecuados. Un cliente MQTT es cualquier dispositivo (desde un teléfono hasta un servidor completo) que ejecuta una biblioteca MQTT y se conecta a un agente MQTT a través de una red.

La información está organizada en una jerarquía de temas. Cuando un editor tiene un nuevo elemento de datos para distribuir, envía un mensaje de control con los datos al corredor conectado. El corredor luego distribuye la información a cualquier cliente que se haya suscrito a ese tema.

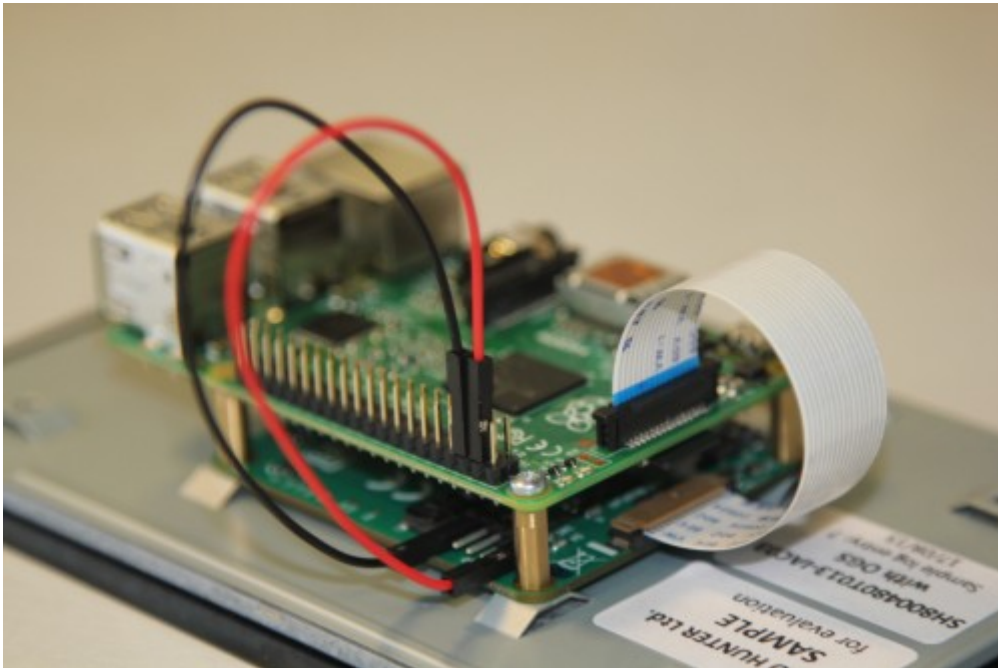
El bróker MQTT es un software que se ejecuta en una computadora (que se ejecuta en las instalaciones o en la nube) y puede ser autoconstruido o alojado por un tercero. Está disponible en implementaciones de código abierto y propietarias.

### **Las principales ventajas del broker MQTT son:**

- Elimina las conexiones de cliente vulnerables e inseguras
- Puede escalar fácilmente de un solo dispositivo a miles
- Administra y rastrea todos los estados de conexión del cliente, incluidas las credenciales y certificados de seguridad
- Reducción de la tensión de la red sin comprometer la seguridad (red celular o satelital)



## Pantalla Táctil RASPBERRY Pi

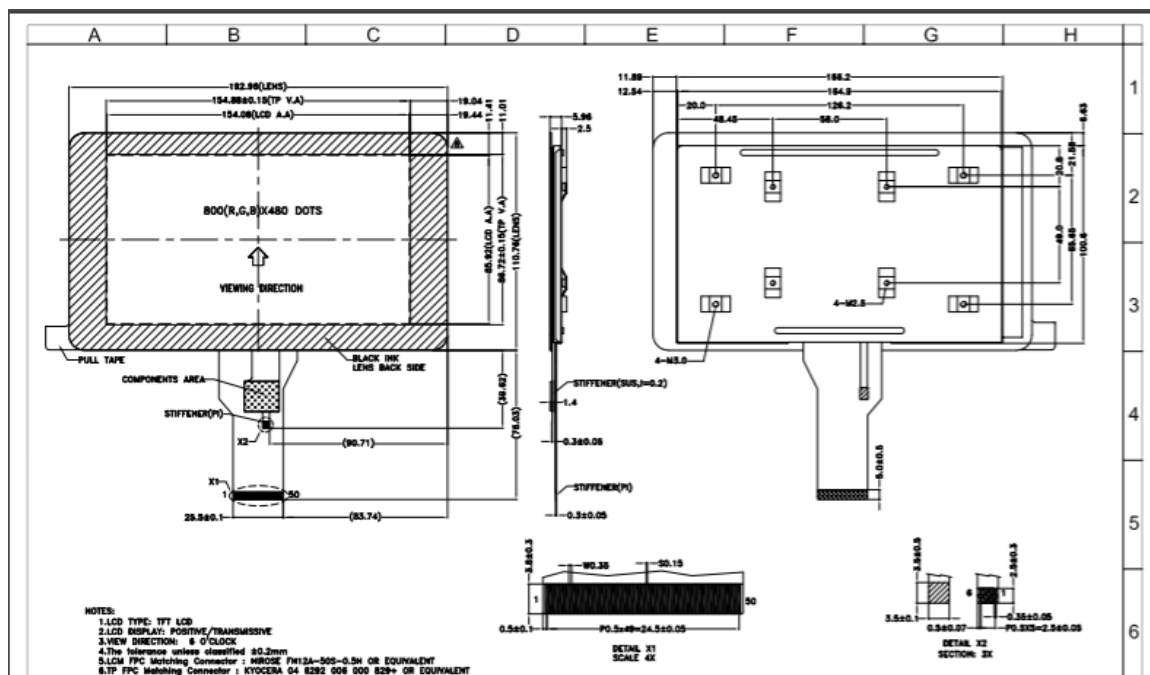


La pantalla táctil Raspberry Pi es una pantalla LCD que se conecta a la Raspberry Pi a través del conector DSI. La pantalla DSI está diseñada para funcionar con todos los modelos de Raspberry Pi, sin embargo, los primeros modelos que no tienen orificios de montaje requerirán hardware de montaje adicional para adaptarse a los separadores dimensionados HAT en la pantalla.

Las pantallas LCD tienen un ángulo de visión óptimo y, dependiendo de cómo se monte la pantalla, puede ser necesario cambiar la orientación de la pantalla para obtener los mejores resultados. De forma predeterminada, la pantalla táctil Raspberry Pi y la Raspberry Pi están configuradas para funcionar mejor cuando se ven desde un poco más arriba, por ejemplo, en un escritorio.

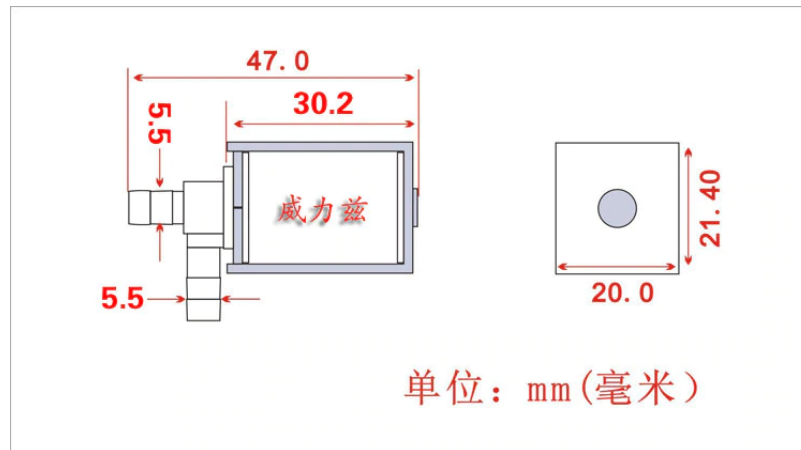
## Especificaciones

- Pantalla LCD RGB de 800 × 480.
- Color de 24 bits.
- Calidad industrial: ángulo de visión de 140 grados horizontal, ángulo de visión de 130 grados vertical.
- Pantalla táctil multitáctil de 10 puntos.
- Control de retroiluminación PWM y control de potencia sobre la interfaz I2C.
- Parte posterior con marco de metal con puntos de montaje para la placa de conversión de pantalla Raspberry Pi y Raspberry Pi.
- Vida útil de la luz de fondo: 20000 horas.
- Temperatura de funcionamiento: -20 a +70 grados centígrados.
- Temperatura de almacenamiento: -30 a +80 grados centígrados.
- Relación de contraste: 500.
- Brillo promedio: 250 cd / m<sup>2</sup>.
- Requisitos de energía: 200 mA a 5 V típico, con brillo máximo.



## Válvulas

Dispositivo que abre o cierra el paso de un fluido por un conducto en una máquina, aparato o instrumento, gracias a un mecanismo, a diferencias de presión, etc.



- Válvula eléctrica de aire y agua de EBOWA, válvula de solenoide Mini DC 6V 12V 24V normalmente cerrada
- Voltaje nominal: DC 6,0 V DC. 12,0 V
- Corriente nominal: <280mA <160mA
- Rango de presión: 0-450 mmHg
- Sellado: la inflación se ha parado después de que la presión de llenado en un contenedor de 500 cc se haya cambiado de 0 a 300 mmHg. Después de la estabilización durante 30 segundos, la presión reducida fue inferior a 5 mmHg/min.
- Fluido de uso: agua/aire
- Vida útil: en un contenedor de 500cc, infle durante 10 segundos y escape durante 5 segundos para probar más de 50.000 veces
- Rango de temperatura de funcionamiento: 0 °C ~ 55 °C
- Potencia: <3w
- Modo de funcionamiento: normalmente cerrado.

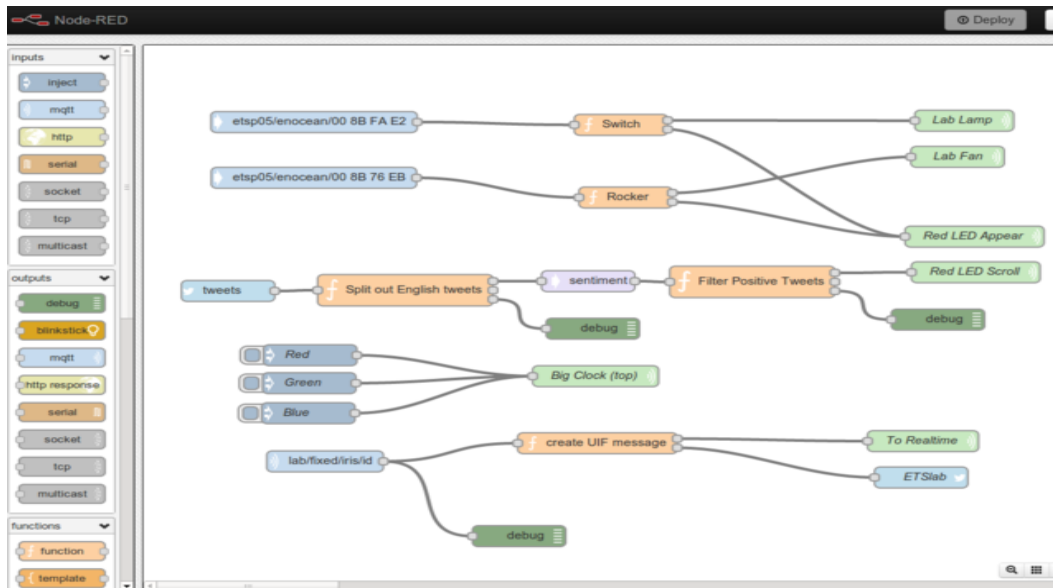
## Node-RED



Node-RED es una herramienta de programación para conectar dispositivos de hardware, API y servicios en línea de formas nuevas e interesantes. Muestra visualmente las relaciones y funciones, y permite al usuario programar sin tener que escribir una línea. Node-RED es un editor de flujo basado en el navegador donde se puede añadir o eliminar nodos y conectarlos entre sí con el fin de hacer que se comuniquen entre ellos.

Node-Red se basa en Node.js. Para instalar el Node-Red, necesitas tener tanto Node.js instalado como NPM. Con NPM, es muy fácil instalar Node-Red: `npm install -g node-red`. NodeJS proporciona la potencia suficiente para que Node-RED sea fiable y escalable. NodeJS es un software muy potente que permite la programación en JavaScript del lado del servidor.

La estructura mínima son los nodos. Estos se arrastran a través de la interfaz gráfica y nos permiten hacer una tarea concreta. Recibir una llamada HTTP, un mensaje MQTT o la activación de un pulsador. Todos estos nodos se organizan en flujos o flows que agrupan nodos que se conectan entre ellos. Todo de una forma visual, sin apenas tener que programar.



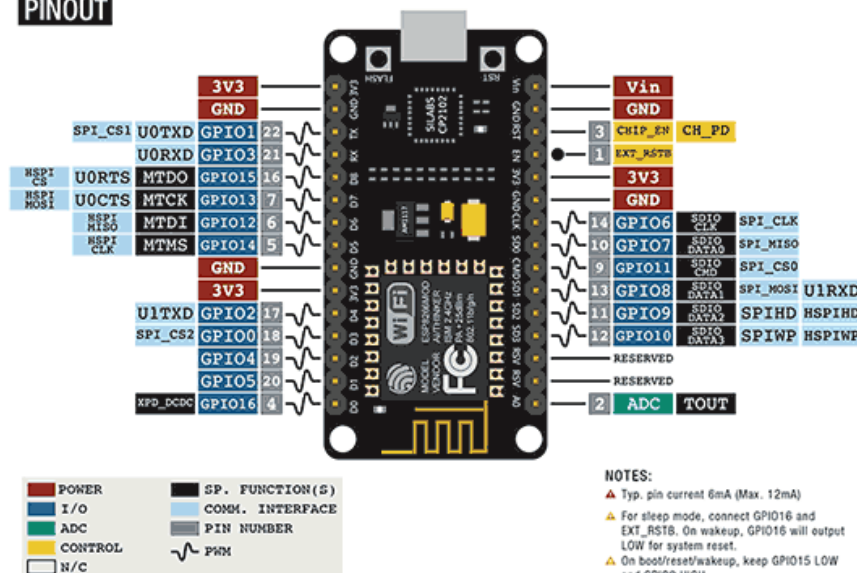
## NodeMCU

NodeMCU es una plataforma IoT de código abierto. Incluye el firmware que se ejecuta en el SoC Wi-Fi ESP8266 de Espressif Systems y el hardware que se basa en el módulo ESP-12. El término "NodeMCU" se refiere al firmware en lugar de a los kits de desarrollo. El firmware utiliza el lenguaje Lua. Se basa en el proyecto eLua y se basa en el SDK no operativo de Espressif para el ESP8266. Utiliza muchos proyectos de código abierto, como lua-cjson, y spiffs.

La placa de desarrollo NodeMCU está basada en el ESP12E y expone las funcionalidades y capacidad del mismo. Pero, además, añade las siguientes ventajas propias de placas de desarrollo:

- Puerto micro USB y conversor Serie-USB.
- Programación sencilla a través del Micro-USB.
- Alimentación a través del USB.
- Terminales (pines) para facilitar la conexión.
- LED y botón de reset integrados.

## NODEMCU V2 PINOUT



# Programa de Arduino

```
#include <SimpleDHT.h>
#include <SimpleDHT.h>
#include <SPI.h>
#define humidity_sensor_pin A0
#define humidity1_sensor1_pin A1
#define humidity2_sensor2_pin A2
#define humidity3_sensor3_pin A3
#define ldr_pin A5

//Bibliotecas para los módulos sensores usados necesarias
//Y definición de variables para los sensores de humedad y LDR en los pines A0 a A5

int pinDHT11 = 12;
SimpleDHT11 dht11;
int ldr_value = 0;

int water_pump_speed = 255;
int water_pump_pin1 = 4;
int water_pump_pin2 = 5;
int water_pump_pin3 = 6;
int water_pump_pin4 = 7;

//Aquí puedes dar valores desde 0 a 255 para la velocidad a la que trabajará la minibomba
//Haz pruebas previas del caudal y configura la. Yo he elegido 255 pero ustedes pueden
elegir la que estimen conveniente. A más velocidad, mayor //bombeo de agua

void setup() {
  Serial.begin(9600);
  Serial.begin(115200);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
```

```

pinMode(5, OUTPUT);

}

void loop() {

    // Mide la temperatura y humedad relativa y muestra resultado
    Serial.println("*****");
    Serial.println("Muestra DHT11...");

    byte temperature = 0;
    byte humidity_in_air = 0;
    byte data[40] = {0};
    if (dht11.read(pinDHT11, &temperature, &humidity_in_air, data)) {
        Serial.print("Lectura del sensor DHT11 fallida");
        return;
    }

    Serial.print("Muestra RAW Bits: ");
    for (int i = 0; i < 40; i++) {
        Serial.print((int)data[i]); if (i > 0 && ((i + 1) % 4) == 0) {
            Serial.print(" ");
        }
    }
    Serial.println("");

    Serial.print("Muestra OK: ");
    Serial.print("Temperatura: "); Serial.print((int)temperature); Serial.print(" *C, ");
    Serial.print("Humedad relativa en aire: "); Serial.print((int)humidity_in_air); Serial.println(
"%");

    int ground_humidity_value = map(analogRead(humidity_sensor_pin), 0, 1023, 100, 0);
    Serial.print("Humedad en suelo: ");
    Serial.print(ground_humidity_value);
    Serial.println("%");

```

```
int ground1_humidity1_value1 = map(analogRead(humidity1_sensor1_pin), 0, 1023, 100, 0);
```

```
Serial.print("Humedad en suelo: ");
```

```
Serial.print(ground1_humidity1_value1);
```

```
Serial.println("%");
```

```
int ground2_humidity2_value2 = map(analogRead(humidity2_sensor2_pin), 0, 1023, 100, 0);
```

```
Serial.print("Humedad en suelo: ");
```

```
Serial.print(ground2_humidity2_value2);
```

```
Serial.println("%");
```

```
int ground3_humidity3_value3 = map(analogRead(humidity3_sensor3_pin), 0, 1023, 100, 0);
```

```
Serial.print("Humedad en suelo: ");
```

```
Serial.print(ground3_humidity3_value3);
```

```
Serial.println("%");
```

```
int ldr_value = map(analogRead(ldr_pin), 1023, 0, 100, 0);
```

```
Serial.print("Luz: ");
```

```
Serial.print(ldr_value);
```

```
Serial.println("%");
```

```
Serial.println("*****");
```

```
//*****
```

```
// Condiciones de riego
```

```
// Si la humedad en el suelo es igual o inferior al 60%, si la luminosidad es inferior al 30%,
```

```
// Si la temperatura es inferior al 35%, entonces el sistema de riego riega.
```

```
// En caso de que no se cumpla alguno o ninguno de los 3 requisitos anteriores,
```

```
// el sistema de riego no riega
```

```
//*****
```

//Aquí puedes variar los parámetros que necesites de 60, 35 y 30, e incluso usar otros operandos <>=...

```

if ( ground_humidity_value <= 50 && ldr_value < 60 && temperature < 25) {
  Serial.println("Regando1");
  digitalWrite(water_pump_pin1, HIGH);
  delay(8000);
}
else {
  digitalWrite(water_pump_pin1, LOW);
  Serial.println("Riego1 detenido");
}

if ( ground1_humidity1_value1 <= 70 && ldr_value < 30 && temperature < 35) {
  Serial.println("Regando2");
  digitalWrite(water_pump_pin2, HIGH);
  delay(8000);
}
else {
  digitalWrite(water_pump_pin2, LOW);
  Serial.println("Riego2 detenido");
}

if ( ground2_humidity2_value2 <= 90 && ldr_value < 40 && temperature < 35) {

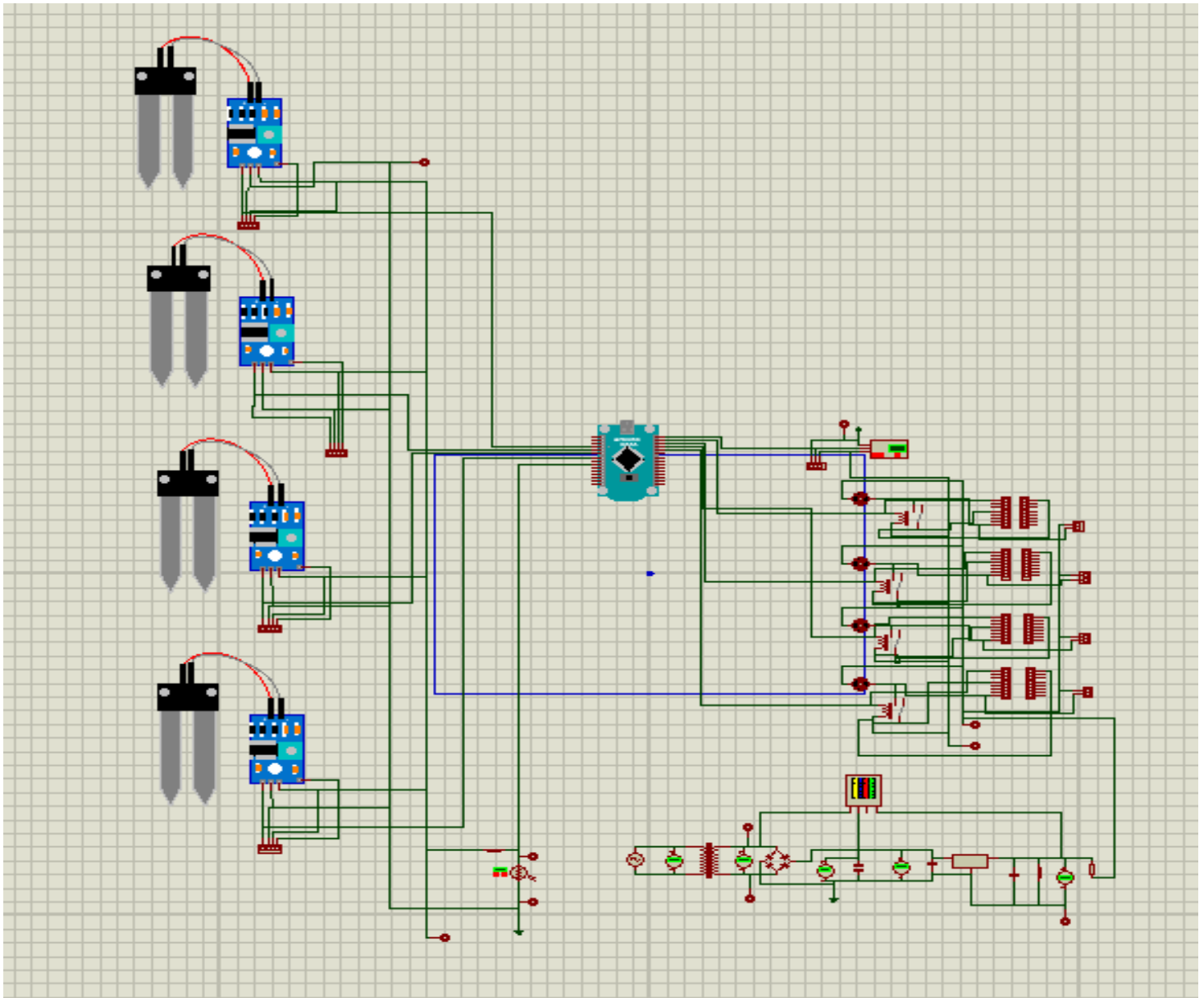
  Serial.println("Regando3");
  analogWrite(water_pump_pin3, HIGH);
  delay(8000);
}
else {
  digitalWrite(water_pump_pin3, LOW);
  Serial.println("Riego3 detenido");
  delay(8000);
}

if ( ground3_humidity3_value3 <= 60 && ldr_value < 30 && temperature < 35) {

```

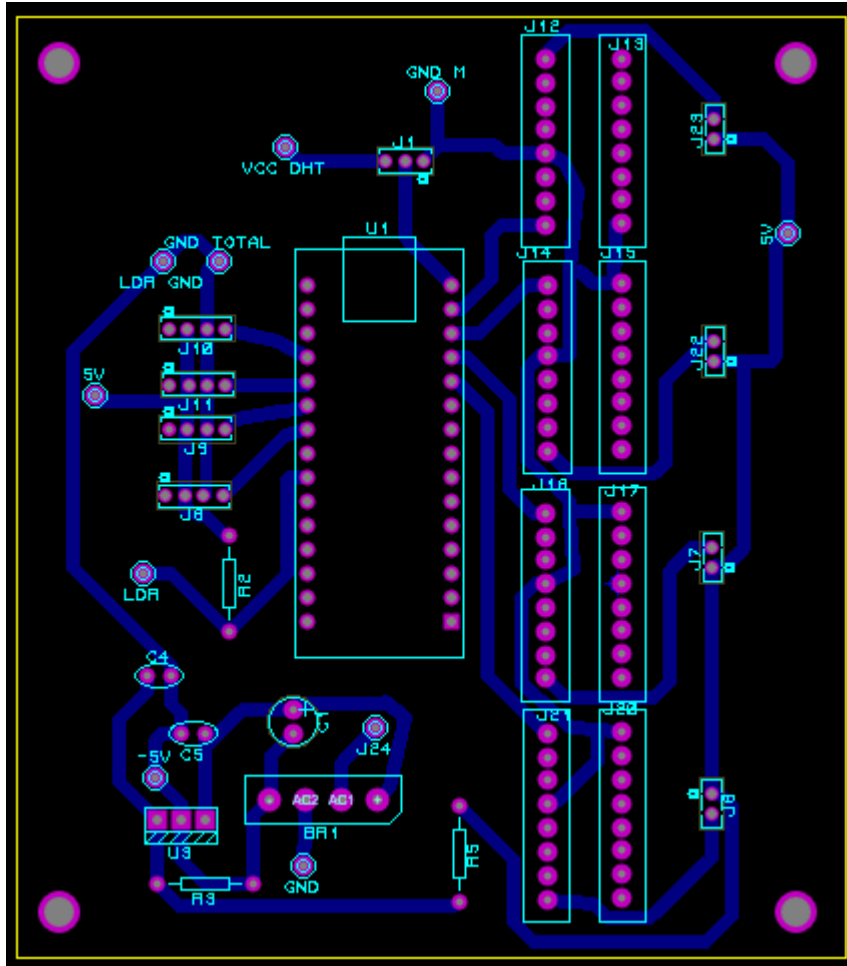
```
Serial.println("Regando4");
digitalWrite (water_pump_pin4, HIGH);
delay(8000);
}
else {
    digitalWrite(water_pump_pin4, LOW);
    Serial.println("Riego detenido4");
}
// Ejecuta el código cada 2000 milisegundos, es decir, 2 segundos. Puedes variar la
frecuencia de muestreo
delay(2000);
}
```

# Esquema eléctrico





## PCB



## Problemas

La parte en la cual tuvimos muchos problemas fue a la hora de crear el programa de arduino y de que se muestre los valores de los sensores en la red (NodeRED).

## Mejoras

Se podría incluir nuevas funciones o hasta poner un reloj esto además de que se pueda controlar de manera remota.

## Conclusión

Para finalizar con nuestro reto nos gustaría comentar nuestra opinión sobre este proyecto. A la hora de empezar nos imaginamos que sería más sencillo pero a medida que nos introducimos en él nos dimos cuenta de que era más complicado de lo que parecía. Hemos aprendido mucho y hemos podido solucionar los problemas por nuestra cuenta, buscando, preguntando y a base de prueba y error.

## Bibliografía

<https://aprendiendoarduino.wordpress.com/category/node-red/>

<https://themicrocontrol.wordpress.com/2017/08/13/node-red-recibiendo-datos-y-controlando-cosas/>

<https://iotdesignpro.com/projects/interface-arduino-with-node-red-to-send-sensor-data-on-webpage>

[https://www.naylampmechatronics.com/blog/34\\_Tutorial-LCD-conectando-tu-arduino-a-un-LCD1.html](https://www.naylampmechatronics.com/blog/34_Tutorial-LCD-conectando-tu-arduino-a-un-LCD1.html)