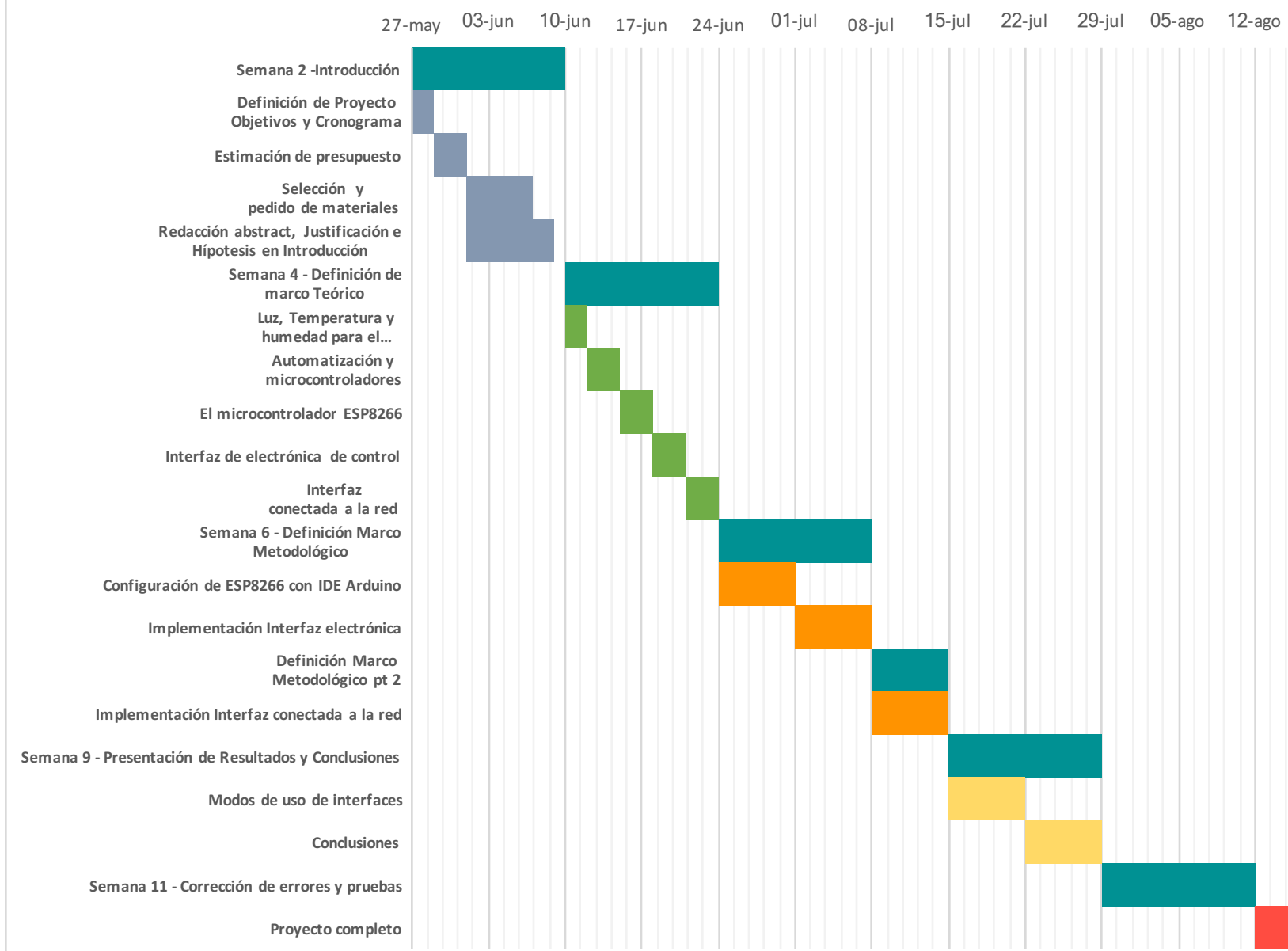


Planeación de Proyecto



Integrantes:

Liliana Rosalba Juárez Polito

Jonathan Pérez Loredó

## **INTERFAZ DE SISTEMA DE CONTROL DE HUMEDAD Y TEMPERATURA IoT**



### **Objetivos Generales**

Diseñar una interfaz de monitoreo y control de las variables de humedad y temperatura de un espécimen de planta. El monitoreo y control se realizará por medio de un dispositivo electrónico que utilizará las variables mencionadas para activar un sistema de riego e iluminación. El usuario será capaz de interactuar con el dispositivo y observar los datos de las variables de control por medio de una interfaz electrónica colocada directamente en la placa del prototipo así como también podrá visualizar y manipular salidas de control por medio de una interfaz visual tipo tablero ya pre configurado a través de un servicio de nube.

### **Objetivos Específicos**

- Implementar un prototipo que sea capaz de obtener lecturas de datos de humedad y temperatura y dar salidas de activación y desactivación de iluminación y riego a un espécimen de planta.

- Interactuar con el dispositivo electrónico por medio de una interfaz electrónica incrustada en la placa la cual controlará y configurará manualmente las salidas de iluminación y de riego de la planta.
- Visualizar las variables de humedad y control así como manipular las salidas del prototipo para riego e iluminación a través de una interfaz conectada a internet.

### **Perfil de usuario**

La intención del prototipo va dirigido a usuarios entusiastas en electrónica, personas que gusten de tener vegetación en el hogar pero que por cuestiones de tiempo no puedan darles el tiempo de cuidado. Personas con intención de iniciarse en procedimientos de huertos automatizados para consumo personal, como entretenimiento o pasatiempo.

## Reporte de avance de prototipo Semana 5.

### 1. Funcionamiento General del Dispositivo

El diagrama funcional del dispositivo muestra la interconexión de diferentes dispositivos que se irán integrando al prototipo controlado por la placa NodeMCU.

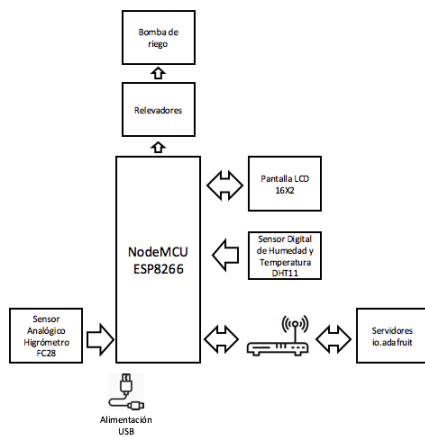


Ilustración 1. Diagrama a bloques del funcionamiento general prototipo. Elaboración propia.

### 2. Conexión de Sensores

- **Sensor higrómetro FC28** se ha conectado a la placa por medio de la entrada conector analógico a digital A0 en la placa del microcontrolador. Este muestra valores entre 0 (máximo nivel de humedad) y 1024 (mínimo nivel de humedad) los cuales se ajustarán de acuerdo al uso del prototipo.

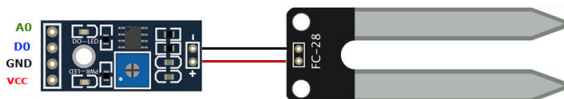


Ilustración 2. Higrómetro FC28

- **Sensor digital DHT11.** Cuenta con un sensor de presión analógico de temperatura y humedad el cual da una salida digital para conexión por protocolo de un cable (1WIRE) que por medio de librerías en el IDE Arduino podremos observar en la consola serial.

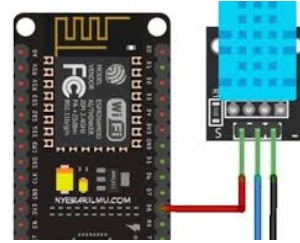


Ilustración 3. Conexión Sensor Humedad y Temperatura DHT11

### 3. Conexión a servidores io.adafruit

“Adafruit.IO es una solución para la construcción de aplicaciones IoT creada por Adafruit Industries, una conocida comercializadora de hardware open-source. Han creado esta plataforma para el internet de las cosas basándose en plataformas conocidas como Arduino, Raspberry pi, ESP8266, Intel Galileo, dispositivos Seriales y Wifi entre otros, La API de comunicación es basado en cliente MQTT con servidores de Adafruit.IO”.

A través del servicio de nube se tiene acceso a un tablero que muestra los datos en tiempo real del prototipo de una conexión WiFi a cualquier servicio de internet disponible.

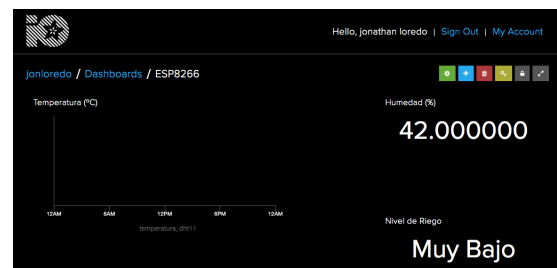


Ilustración 4. Tablero io.adafruit de datos recolectados de la placa

## Reporte de avance de prototipo Semana 7.

### 4. Interfaz electrónica para visualizar datos por medio de display LCD 16X2

La visualización de datos obtenidos de los sensores es posible realizarla mediante una interfaz LCD 16X2. La conexión del dispositivo LCD se comunicará con la placa NodeMCU por medio de un extensor de conexiones o demultiplexor PCF8574 ya integrado en una placa lista para comunicación I2C usando pines SCL y SDA en placa microcontrolador. La configuración en el código Arduino se hace mediante el uso de la librería LiquidCrystal\_I2C.h la cual está hecha especialmente para realizar la comunicación I2C con el dispositivo

### 5. Ajuste de sonda de prueba de Higrómetro FC28 al puerto A0.

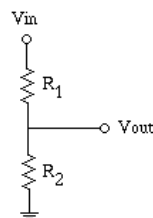
La conexión de la sonda de prueba a la placa NodeMCU se hace utilizando el pin ADC A0. Este cuenta con una resolución de 10 bits escalando valores de voltaje que oscilan entre 0V y 1V. Como medida de protección para la placa, el fabricante ha incluido un divisor de voltaje para asegurarse que la tensión máxima a la entrada A0 sea máximo 1V.

A su vez, con la intención de contar con una mejor referencia sobre la cual medir el voltaje que muestra el nivel de riego. Se realiza la conexión de un potenciómetro con valor de 10KΩ conectado en serie con la sonda de prueba la cual actúa también como un resistor variable. Esta conexión de divisor de voltaje de sonda de prueba y potenciómetro ahora sí contará con una interfaz para ajustar valores de nivel de

riego antes de pasar a la entrada A0 de la placa del microcontrolador la cual internamente cuenta con un divisor de voltaje asignado de la siguiente manera:

Voltage Divider

Para:



$$R1 = 220K\Omega$$

$$R2 = 100K\Omega$$

$$V_{out} = \frac{R_2}{R_1 + R_2} V_{in}$$

### 6. Esquemático de conexiones

El diagrama eléctrico como se observa incluye conexiones mínimas a circuitos externos.

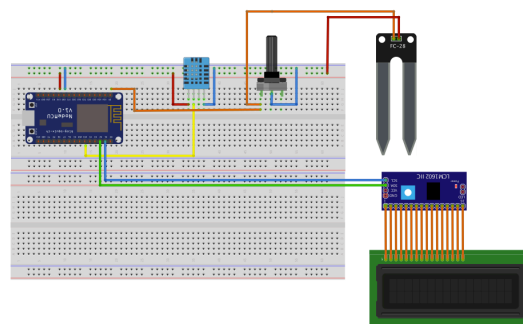


Ilustración 5. Esquemático de conexiones

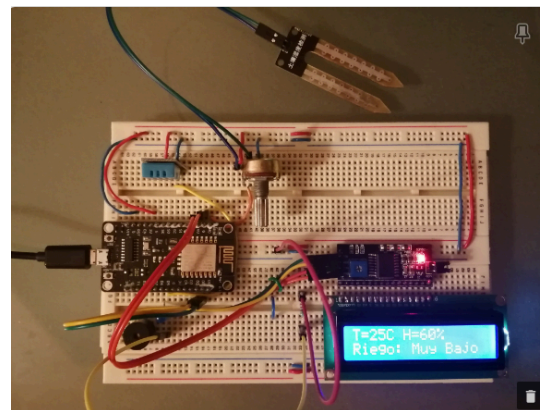


Ilustración 6. Muestra prototipo

## Reporte de avance de prototipo Semana 9.

### 7 Uso de librería Ticker.h para ejecución de funciones en segundo plano.

Mientras se trabaja con el programa principal del prototipo se nota que no es posible utilizar ciertas funciones durante la ejecución de comando `delay()`. Es necesario solucionarlo mediante el uso de un pseudo-temporizador utilizando la librería `Ticker.h`. Esto permite realizar la ejecución del envío de datos a los servidores de `adafruit.io` para actualizar los valores de temperatura y humedad cada 30 segundos.

```
/*Se utiliza Ticker como pseudo-temporizador cada 30 segundos llamando la
función nivel_riego() con operador attach para poder seguir haciendo llamadas a otras funciones
el programa ej enlace https://circuitstoday.com/2018/01/02/esp8266-timer-ticker-example/ */
tiempo1.attach(30,nivel_riego);

//Función para enviar valores de nivel de riego, temperatura y humedad a Adafruit

void nivel_riego(){

    temperatura2io->save(temperatura_dht11);
    humedad2io->save(humedad_dht11);
    nrriegosensor->save(n_riego);

}
```

### 8 Configuración de señales de entrada a placa NodeMCU.

La placa NodeMCU cuenta con una configuración de fábrica para ciertos pines de entrada. Para el uso de este prototipo se ha escogido la entrada GPIO2 como selección para activar una función MANUAL/AUTOMATICO que responderá a un voltaje enviado por medio de un botón pulsador. Internamente la entrada GPIO2 se encuentra conectada con una resistencia de PULL-UP (referencia:

<https://tttapa.github.io/ESP8266/Chap04%20-%20Microcontroller.html>

) por lo cual la señal se encuentra “invertida” siempre en estado HIGH o alto y al momento de pulsar el botón (el cuál es necesario conectarlo a tierra) dará un voltaje LOW.

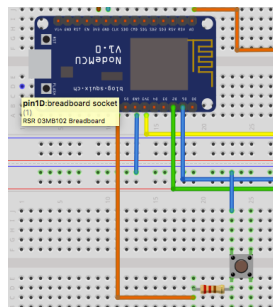


Ilustración 7. Botón pulsador colocado a una entrada GPIO2 por default PULL UP

### 9 Integración modo AUTOMATICO/MANUAL

La funcionalidad del prototipo termina con una opción para colocar el dispositivo en modo automático o manual. Esto es necesario ya que para el uso del mismo, el usuario tendrá acceso a una interfaz por medio de la placa NodeMCU con un botón pulsador para activar dicha función. Esta activará un indicador de sonido para hacer saber al usuario si ha sido encendida, también enviará una notificación a los servidores de `adafruit.io` para hacerle saber su estado y determinar si la función AUTOMATICO/MANUAL ha sido usada y alternará el indicar en los servidores tanto en la placa para poder usarla sin bloquear su estado anterior, es decir, si la placa tenía AUTOMATICO encendido se verá activado tanto en la placa como en los servidores `adafruit.io`.

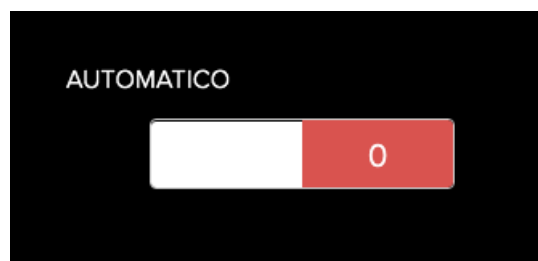


Ilustración 8. Modo AUTOMATICO/MANUAL