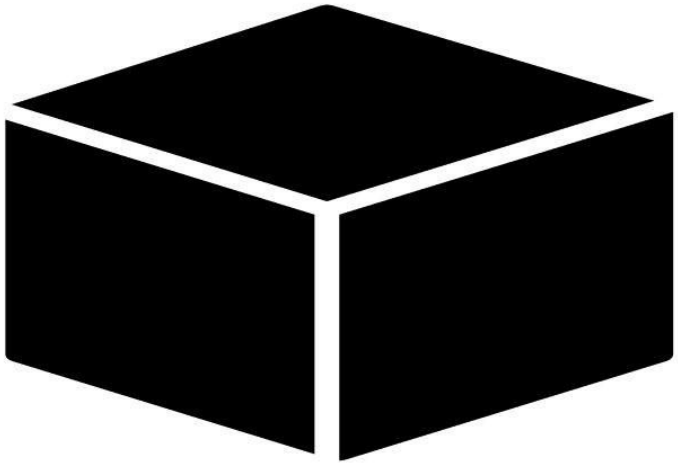
# SONDA AGRÍCOLA
# DOCUMENTACIÓN



SensorTech

Abidán Brito Clavijo
Pablo Enguix Llopis
Luis Belloch Martínez

CDIO

Elvira Montagud Hernandis

# HISTORIA DE REVISIONES

| Versión | Novedades | Fecha |
|---|---|---|
| Sprint 3 | Escalabilidad<br>Sensor de Luz<br>GPS<br>Acelerómetro<br>Presión atmosférica | 09/12/2019 |
| Sprint 2 | Librería Sensores.h<br>Sensor de temperatura<br>Modo Deep Sleep | 18/11/2019 |
| Sprint 1 | Sensores de humedad y salinidad | 28/10/2019 |

# ÍNDICE DE CONTENIDOS

REPOSITORIO GIT Y LISTADO DOCUMENTAL 3

DISEÑO DEL SISTEMA 3

ESQUEMÁTICOS ELECTRÓNICOS 6

DISEÑO DEL PROGRAMA 10

CALIBRACIONES Y TESTEOS 11

DIAGRAMA DE BURNDOWN 14

DAILY SCRUM 14

CDIO

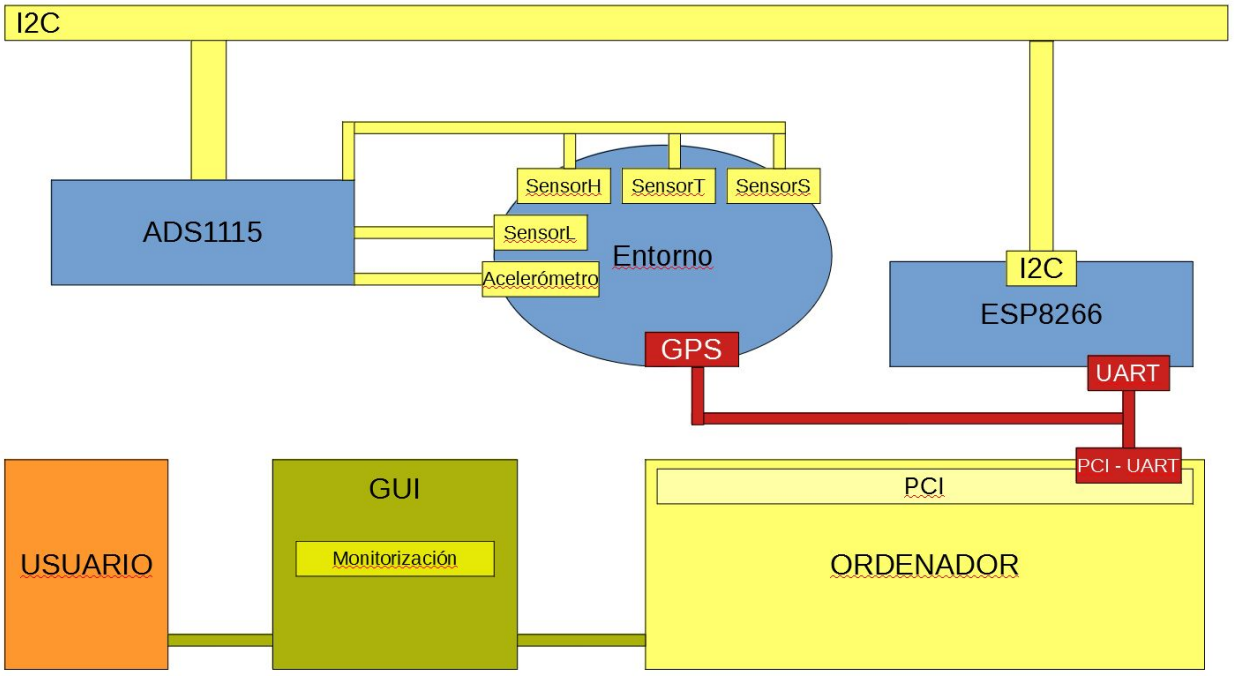
# REPOSITORIO GIT Y LISTADO DOCUMENTAL

Para la base de código y documentos pertinentes de este proyecto, se ha utilizado un repositorio Git como sistema de control de versiones, alojado en la plataforma GitHub. Para acceder al repositorio, haga click sobre el hipervínculo aquí expuesto: https://github.com/abidanBrito/CDIO_Agriculture_Sensors

Todos los documentos de esta versión se encuentran disponibles tanto en la aplicación de gestión de proyectos ágiles Worki, como en el repositorio de GitHub arriba citado. El repositorio está compuesto por los siguientes documentos y directorios:

1. **Sprint2_Documentación.pdf.** (Este documento.)
2. **Actas_Daily_Scrum.pdf.** (Actas de las reuniones.)

# DISEÑO DEL SISTEMA

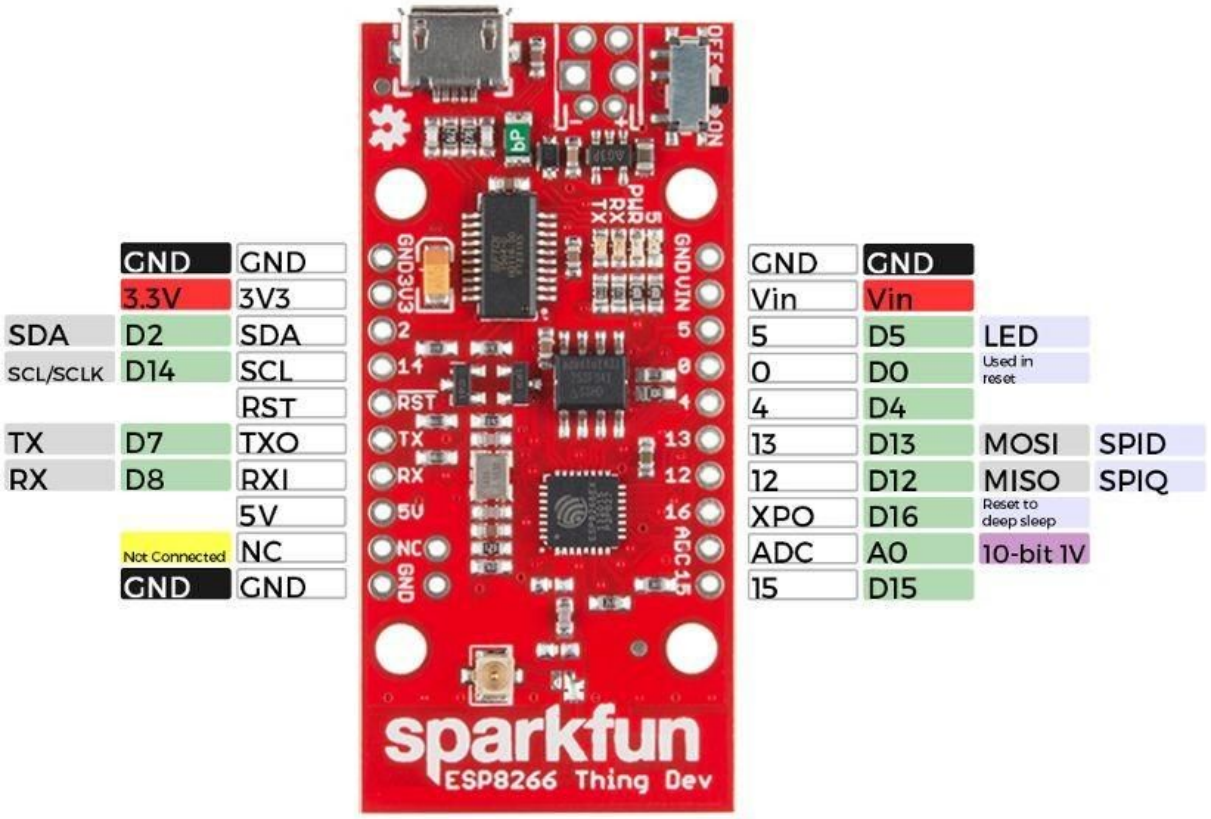



**COMPONENTES, BUSES Y PINES RELEVANTES**

El sistema aquí expuesto parte del microcontrolador **ESP8266 Thing Dev** como base, acoplado al **convertidor Analógico-Digital** (A-D / ADC) **ADS1115**. Es necesario el uso de un ADC externo para poder leer / registrar varias señales analógicas a la vez. Por ende, los sensores se conectan al ADC. El

## CDIO

ADS1115 dispone de 4 ADC de **16 bits**, 15 para la medición y 1 para el signo. A modo de inciso, cabe mencionar que la resolución del ADC se ajusta al rango de entrada.




Lo primero que se debe hacer es establecer una línea de tierra. Los pines de **GND** han de coincidir, y se ha de tener especial precaución con el sentido en que se acoplan los pines. Si se hiciese en una orientación errónea, se podría cortocircuitar la placa.

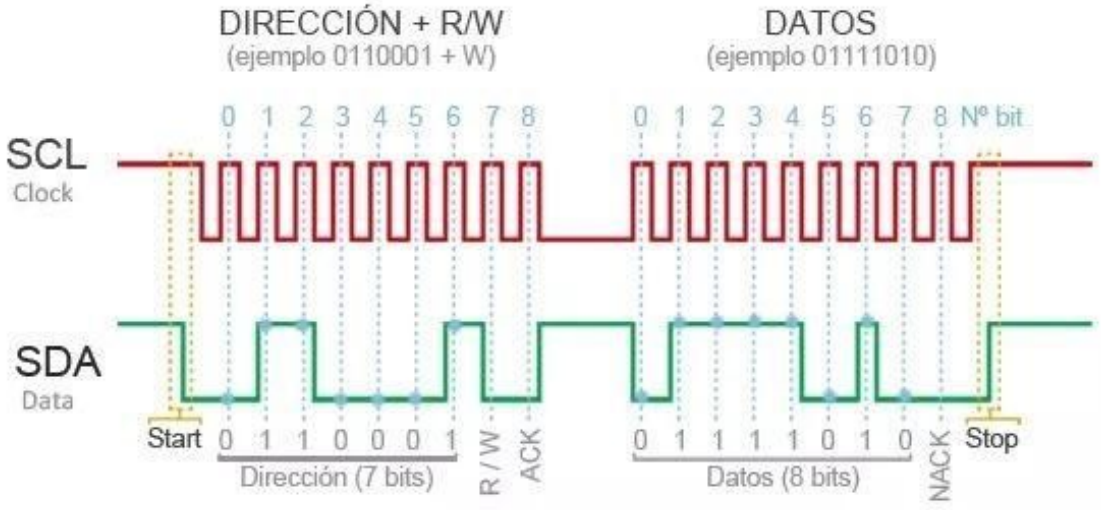
El circuito del **Transmisor-Receptor Asíncrono Universal (UART)** controla los puertos y dispositivos serie. Se conecta al ordenador, mediante un protocolo de comunicación en serie, permitiendo el análisis en tiempo real con el monitor en serie del Arduino IDE. La velocidad de transferencia ha de ser la misma que la configurada en el Arduino, para que la transmisión de datos coincida. Nótese que la placa ESP8266 solo tiene un UART. Es decir, no pueden haber varios periféricos conectados al mismo tiempo por la misma UART. La transmisión se lleva a cabo por el pin **TRx**. No obstante, al conectar la placa al PC por el puerto **microUSB**, se conecta directamente al UART.

Nuestro sistema utiliza el **protocolo I2C.** Se trata de un protocolo de comunicación en serie, bidireccional (half-duplex) y síncrono. El **bus I2C** se conecta a todos los sensores. Es el mismo para todos. La conexión al microcontrolador se lleva a cabo a través de dos líneas o buses:

- **SDA** ( data): primero se manda el bit de inicio (start) y luego la secuencia; el tren de datos en sí. Finalmente, un bit de acuerdo y otro de parada.
- **SCL / CLK** (clock): bus para el reloj. Marca el ritmo o compás, por así decirlo.

### CDIO

Ésto quiere decir que el maestro y el esclavo envían datos por el mismo bus, el cual es controlado por el maestro, que crea la señal de reloj. Nótese que I2C no utiliza selección de esclavo, sino direccionamiento.




Las líneas SCL del microcontrolador y SCL del convertidor Analógico-Digital se conectan entre sí, sincronizándolos. Análogamente, los SDA y GND se conectan por pares. En nuestro caso la facultad nos ha proporcionado los componentes, y el ADC consta de un circuito integrado que redirecciona las conexiones alineando los pines, facilitando el acoplamiento.

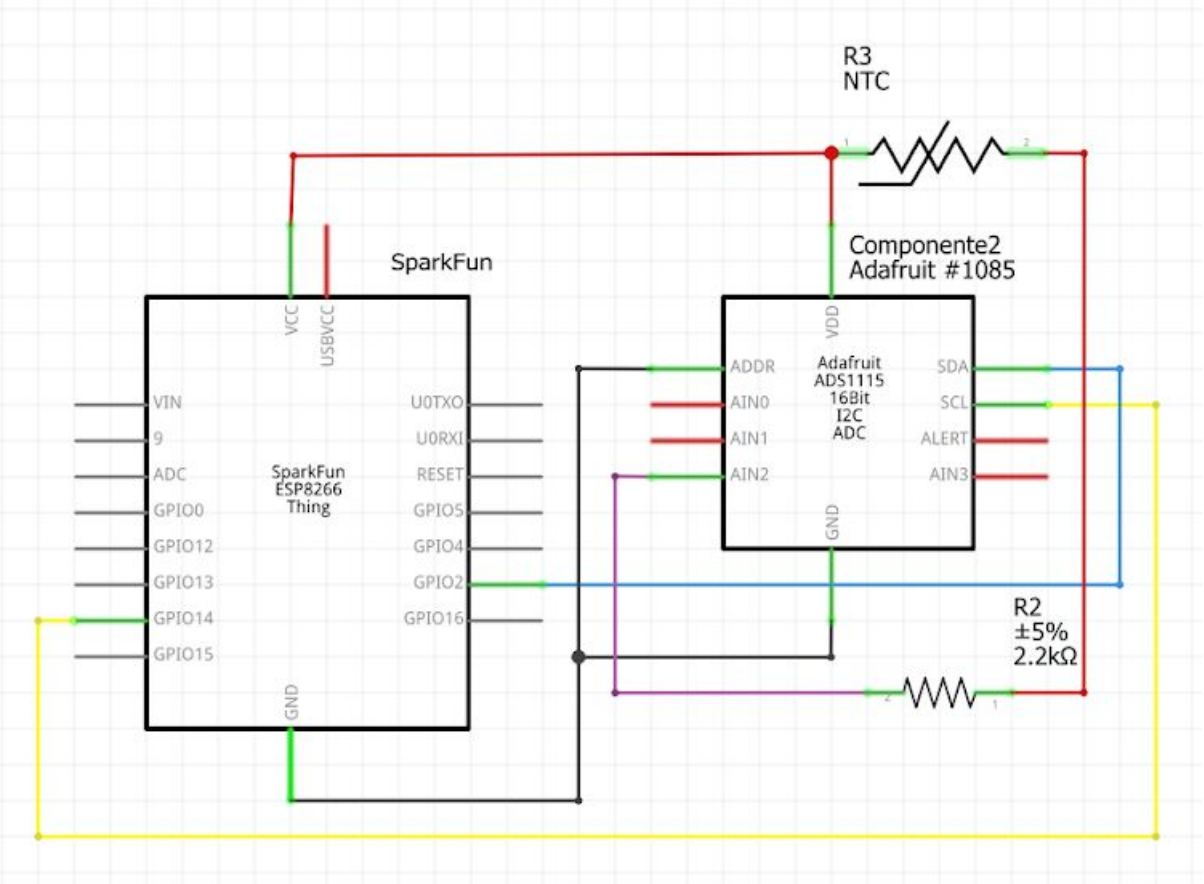
## ESQUEMÁTICOS ELECTRÓNICOS

Para la elaboración de los esquemáticos electrónicos se ha utilizado el programa informático *Fritzing;* una herramienta especializada para la creación de esquemáticos, circuitos electrónicos y diseño de PCBs.

Seguidamente se presenta el esquema del sensor de temperatura NTC, por ser el añadido con respecto al sprint anterior, y el esquema del montaje de la función *Deep Sleep*.

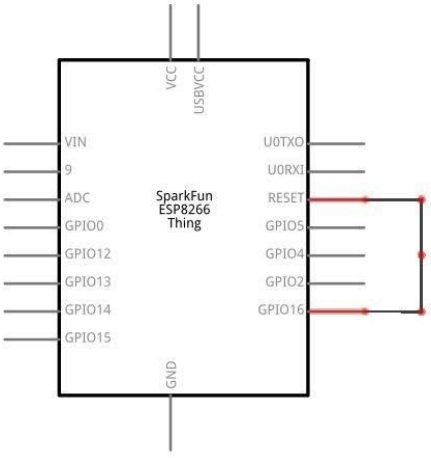
# CDIO

**SENSOR DE TEMPERATURA**




Véase el archivo original adjunto en el directorio de GIT Hub *"Sprint2/schematics/SensorTemperatura.fzz"* en el repositorio o en la sección de documentos de la aplicación de gestión de proyectos ágiles Worki.

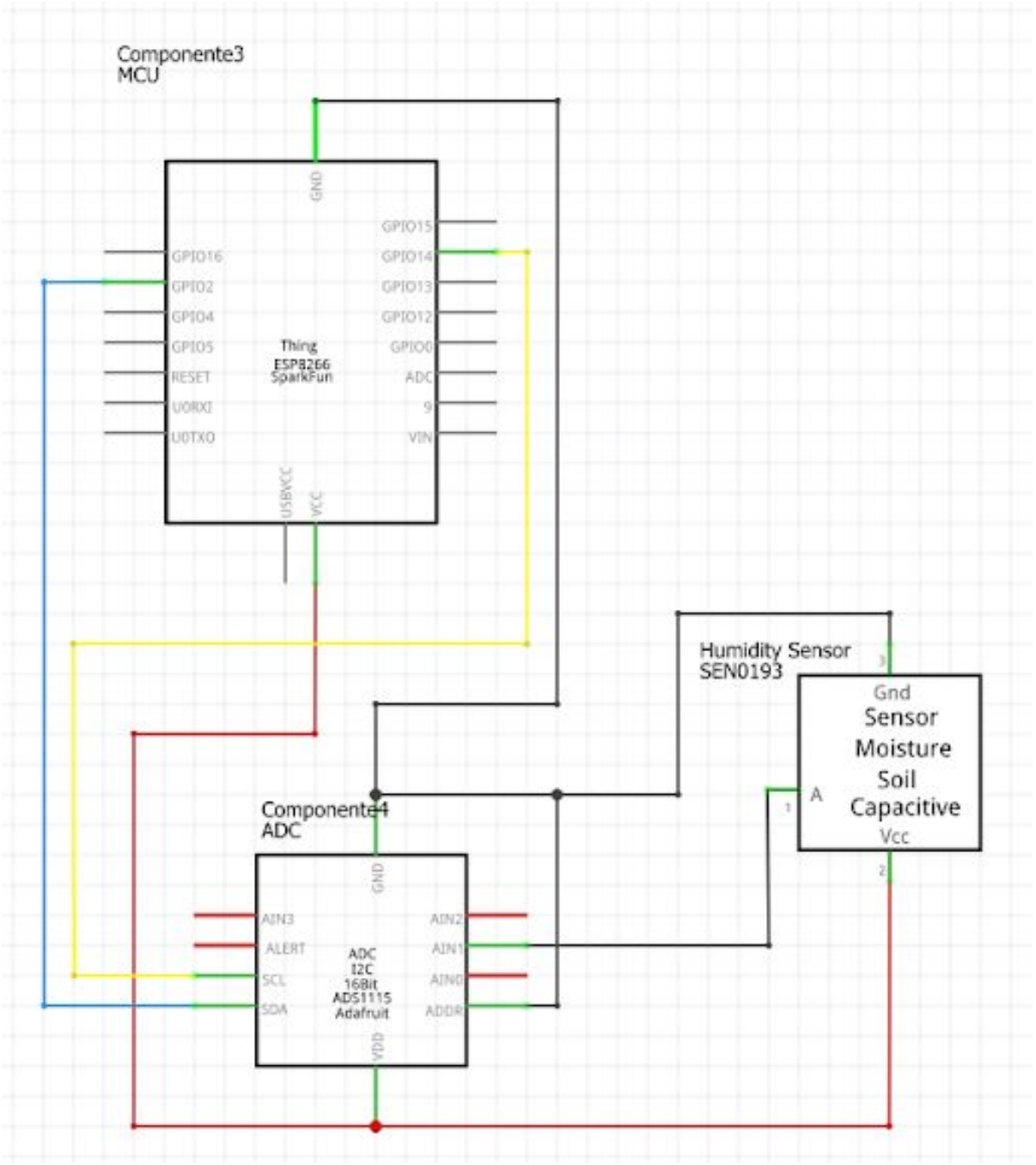
**DEEP SLEEP**



## CDIO

El archivo original se encuentra en el repositorio GIT "/Sprint2/schematics/DeepSleep.fzz".

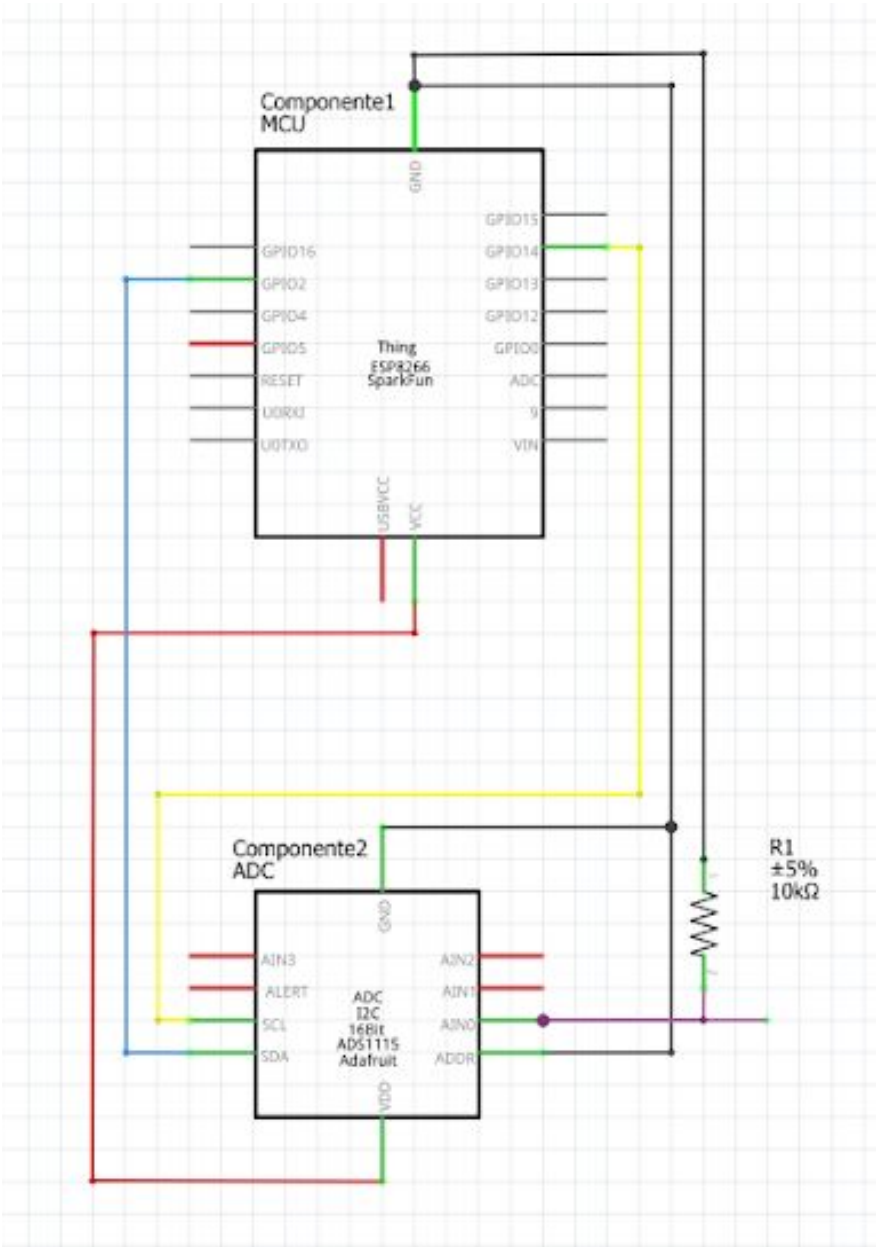
**SENSOR DE HUMEDAD**




Véase el archivo original adjunto en el directorio *"Sprint1\schemes\SensorHumedad.fzz"* en el repositorio o en la sección de documentos de la aplicación de gestión de proyectos ágiles Worki.

# CDIO

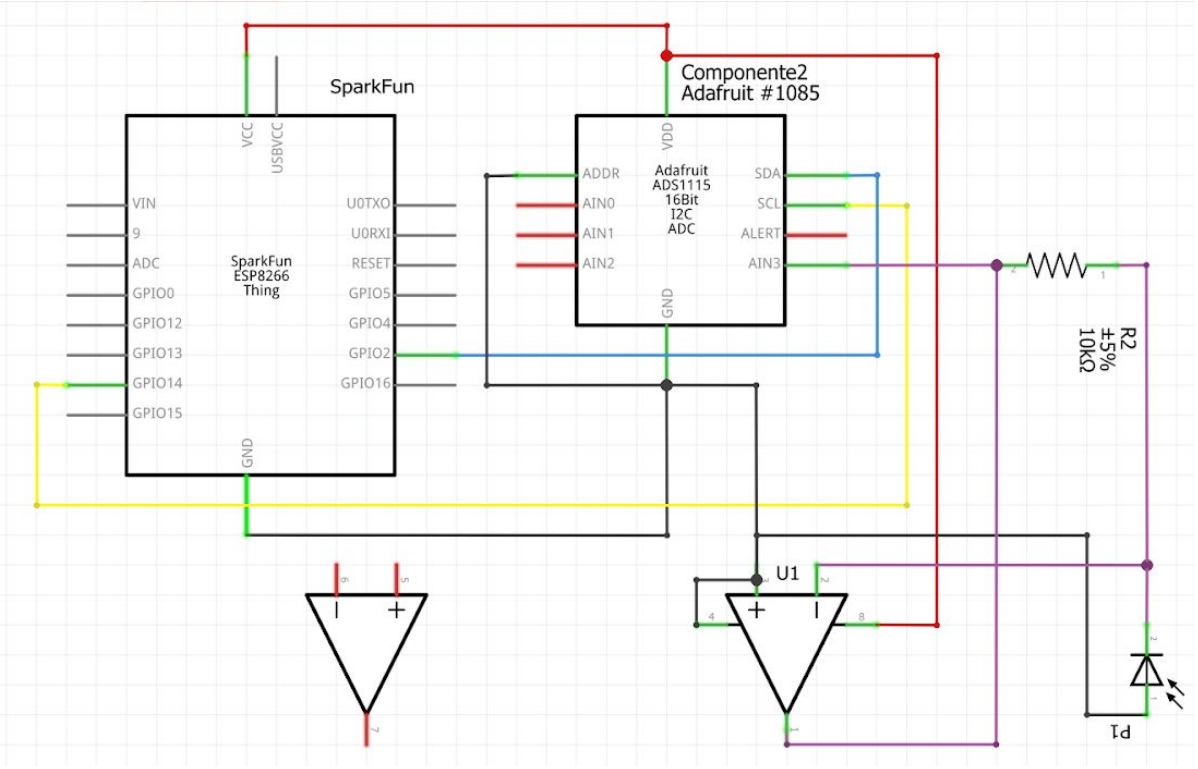
**SENSOR DE SALINIDAD**



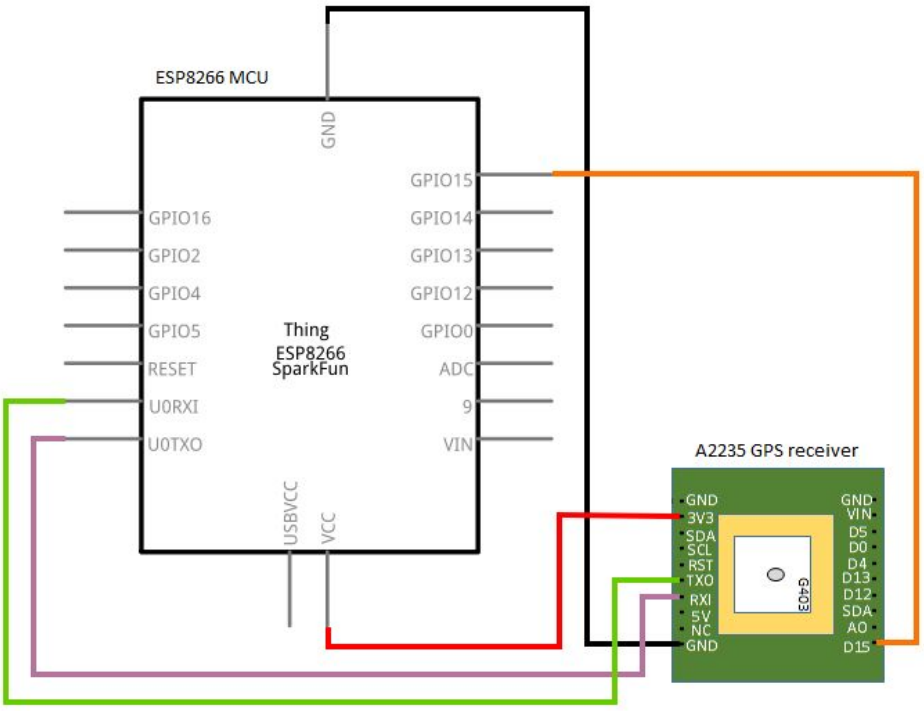

Véase el archivo original adjunto en el directorio *"Sprint1\schemes\SensorSalinidad.fzz"* en el repositorio o en la sección de documentos de la aplicación de gestión de proyectos ágiles Worki.

## CDIO

**SENSOR LUZ**

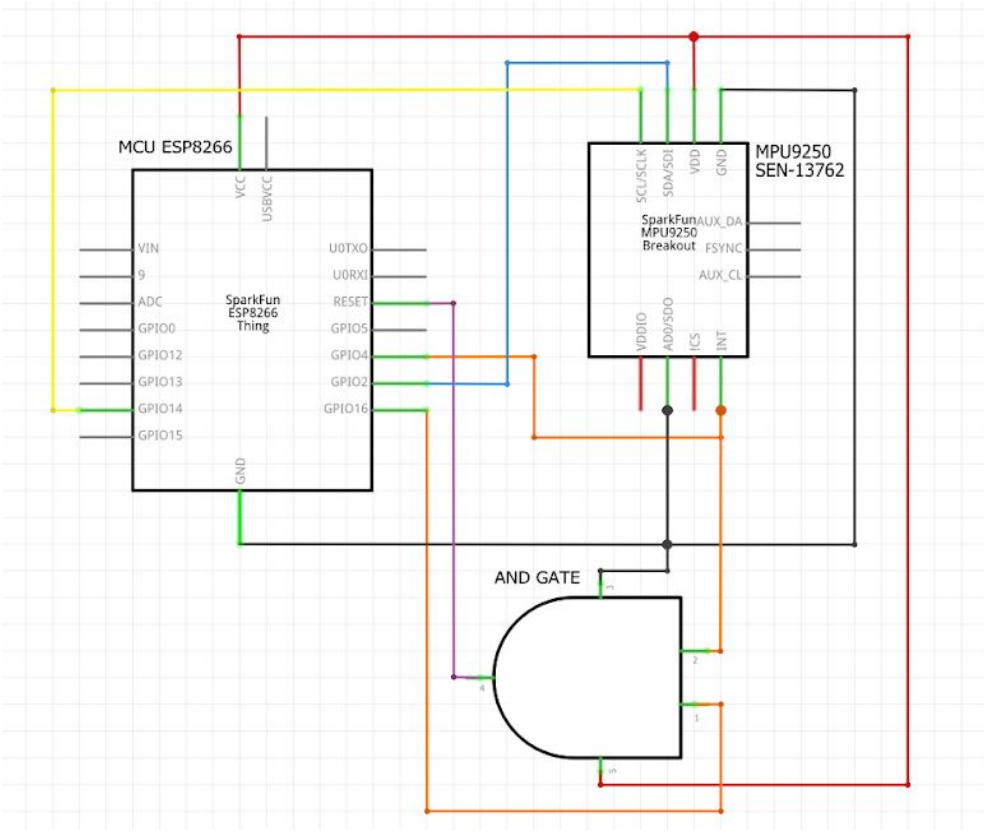



**GPS**



## CDIO

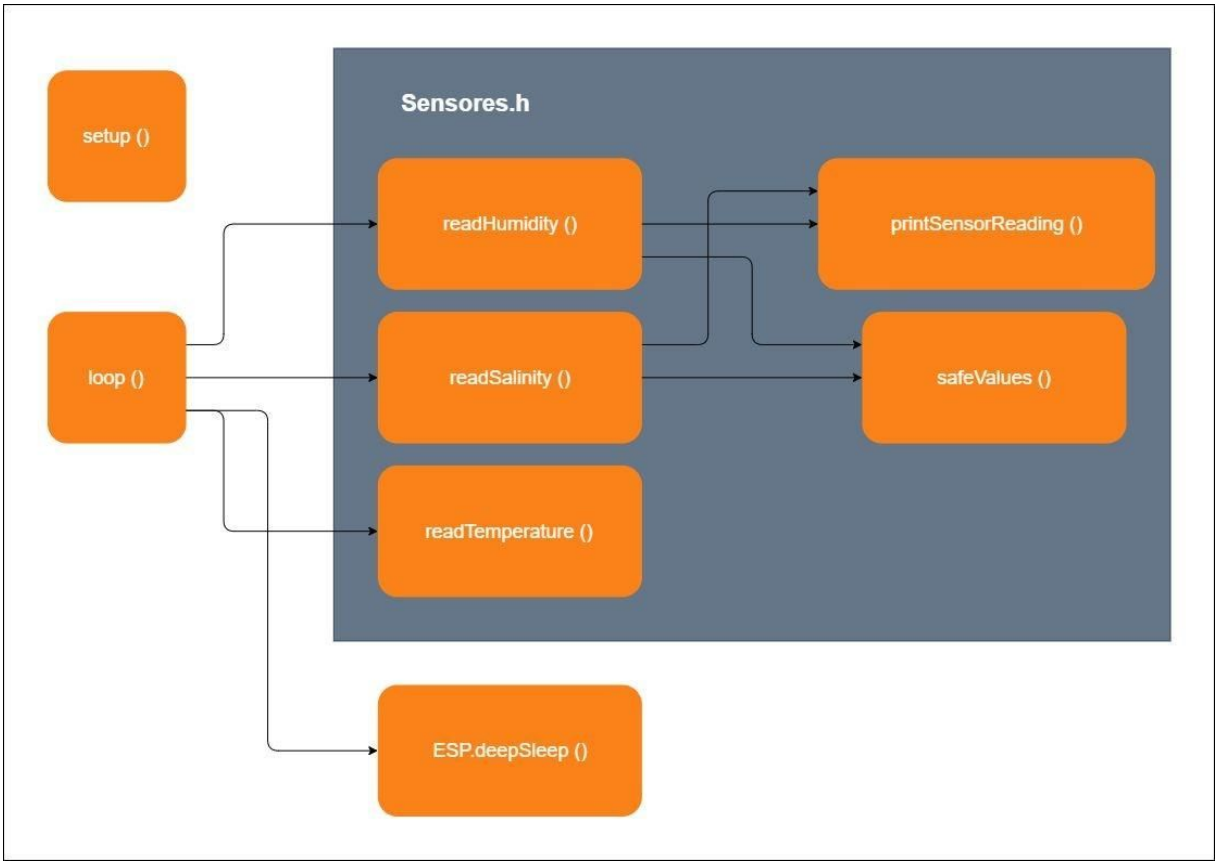
**ACELERÓMETRO**




# DISEÑO DEL PROGRAMA

Se adjunta a continuación el esquema de la librería Sensores.h, la cual agrupa las funciones de todos los sensores con el fin de simplificar el código.

## CDIO




El archivo original se encuentra en "*/Sprint2/img/softwareArchitecture.png*".

En este diseño se presentan las funciones de los tres sensores implementados hasta este sprint: salinidad, humedad y temperatura. Además se incluye una función que imprime los resultados recibidos en el monitor en serie del Arduino IDE, así como otra función llamada "safeValues" que evita que éstos salgan del rango establecido (0 - 100).

Esta librería es accedida desde *mainProgram.ino* el cual contiene dos funciones:

- **void Setup()**: se ejecuta una vez y define parámetros iniciales como la ganancia del ADS.
- **void Loop()**: hace las llamadas a las funciones de Sensores.h y guarda algunos datos. Se ejecuta cada minuto con la función Deep Sleep (configurable).

Para poder utilizar el ADC externo, es necesario:

- Añadir la **librería** correspondiente.
- Definir el **objeto** ADC.
- Inicializar el ADC (con la función **miADC.begin ()**).
- Definir la **ganancia** con la que se va a trabajar (al inicio del programa).

## CDIO

El ADC sólo funciona en el **rango (0, 1) V**. No obstante, podemos utilizar la función **map()** de Arduino para transformar / escalar los valores de un rango de entrada a uno de salida. Es importante asignar un valor de **ADDR**, para determinar la dirección (valor

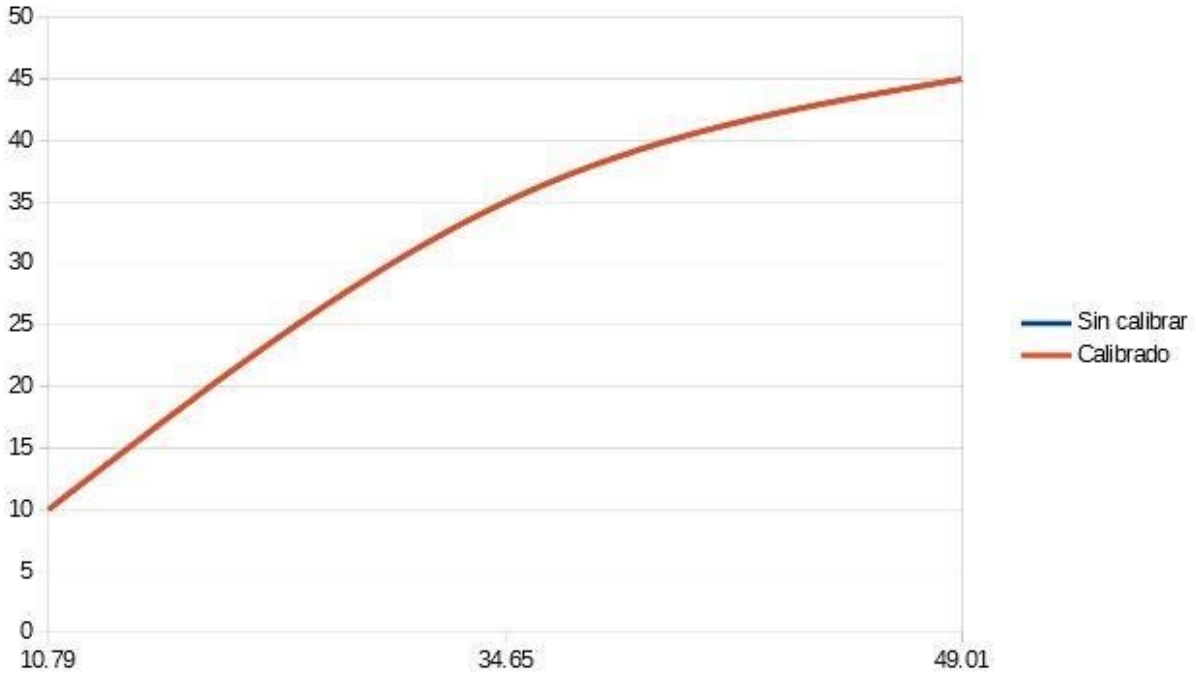
hexadecimal) en función del pin. Ésto es necesario para discernir qué convertidor analógico utilizar.

*Sensores.h* y *mainProgram.ino* constituyen el programa completo que se puede encontrar en el repositorio GIT: "*CDIO-SensorTech/Sprint2/ code/mainProgram/*"

# CALIBRACIONES Y TESTEOS

### SENSOR DE TEMPERATURA

Para calibrar el sensor de temperatura hay que tener en cuenta que el valor de las temperaturas solo es óptimo en temperaturas que oscilan entre los 0ºC y los 45ºC, por ello en la calibración de este sensor se han usado como base un vaso con agua a temperatura ambiente y otro vaso de agua con un poco de hielo, para así realizar con la mayor precisión el calibrado. En la fase de calibrado es imprescindible el uso de un termómetro para así asignar a los valores de la NTC a los valores medidos con este. Los datos recibidos se ajustan con la variación media con respecto a la recta de regresión.

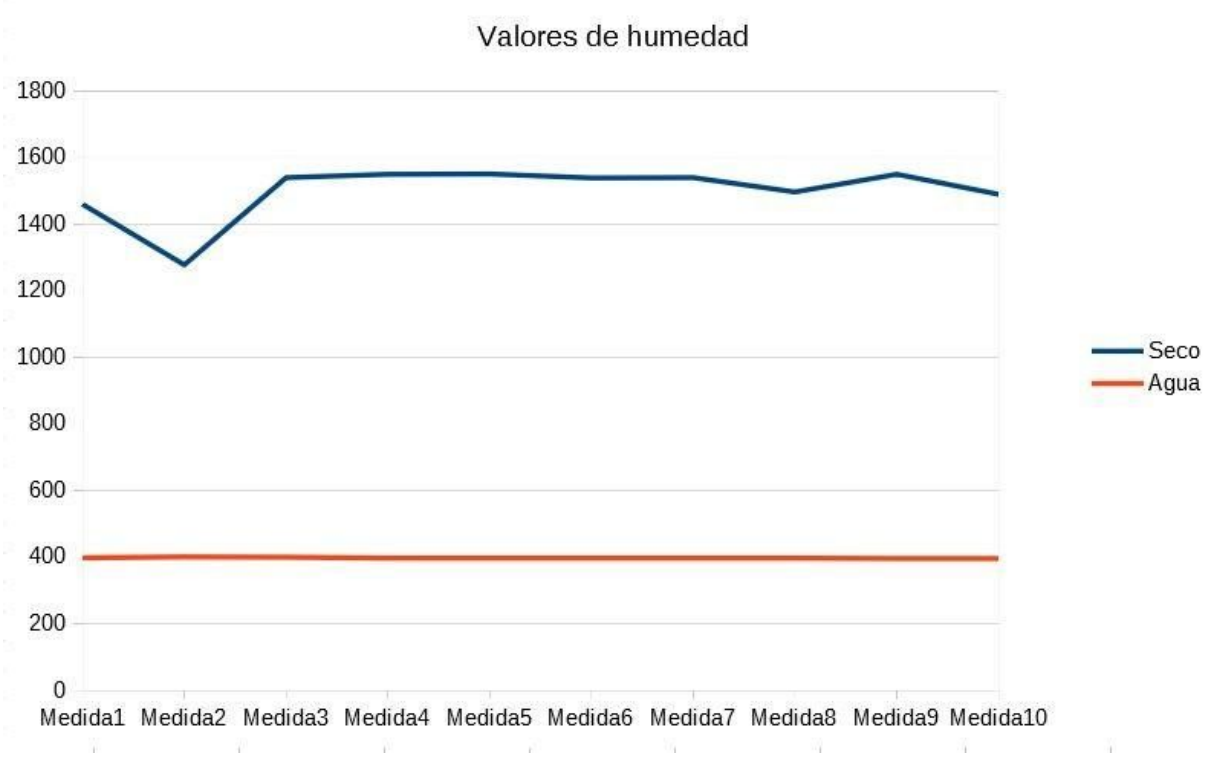



Podemos ver que la desviación es de -0.24ºC de media, por ello el cambio es casi imperceptible. El sensor es suficientemente preciso para este proyecto.

### SENSOR DE HUMEDAD

Para calibrar el sensor de humedad se toman como punto de referencia dos medios (el aire y el agua). Es decir, realizaremos mediciones con nuestro sensor en un medio seco (aire), y uno acuoso

(agua). Se hará un promedio de éstas y se establecerán niveles de humedad cercanos al 0% y al 100%, respectivamente. Finalmente, en la fase de testeo, hacemos mediciones en distintos medios y bajo diferentes condiciones (seco, mojado, parcialmente mojado, húmedo, etc.), para comprobar el correcto funcionamiento del sensor en función a las fluctuaciones del promedio recogidas.




Véase una tabla con las mediciones realizadas durante la calibración del sensor.

**SENSOR DE SALINIDAD**

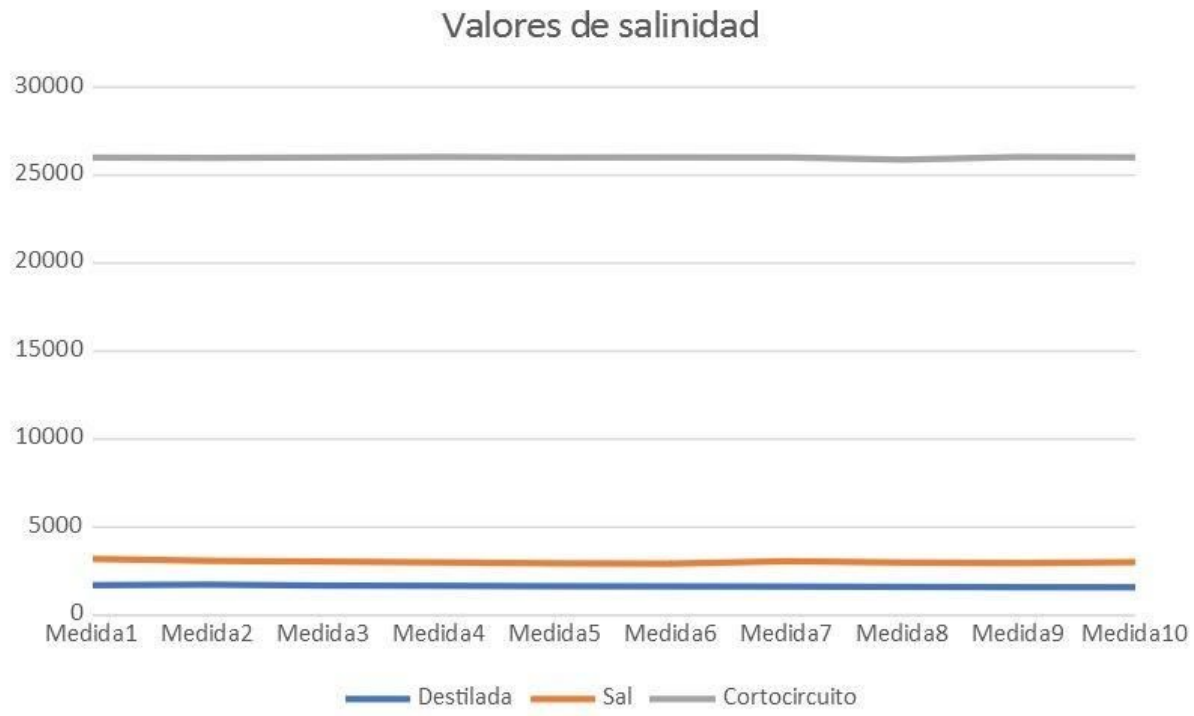
Para calibrar el sensor de salinidad es importante comprender la conductividad eléctrica en medios líquidos. A aquellos líquidos que, por la presencia de iones libres en su

composición, se convierten en conductores eléctricos se les conoce como electrolitos. Las soluciones salinas son electrolitos. Los terminales eléctricamente cargados que se introducen en la solución acuosa se denominan electrodos. El terminal con carga positiva es el ánodo. El terminal con carga negativa es el cátodo. De esta manera, analizaremos la conductividad eléctrica entre electrodos en diferentes soluciones para calibrar nuestro sensor y minimizar el error en la medida (estableciendo un límite superior y uno inferior).

En primer lugar, tomaremos medidas en agua destilada (no conductora), y ajustaremos el rango tal que el promedio de los valores represente un nivel de salinidad cercano al 0%. En segundo lugar, tomaremos medidas con los electrodos en contacto (circuito cerrado), siendo así la conductividad eléctrica máxima, ajustando el rango tal que el promedio representa un nivel de salinidad cercano al 100%. Acrecentamos ligeramente los límites en opuestas direcciones; como margen de error, evitando así lecturas negativas al realizar promedios y transformar los valores de un rango a otro. Finalmente, en la fase de testeo, hacemos mediciones en agua con diferentes cantidades de sal para

comprobar el correcto funcionamiento del sensor en función a las fluctuaciones del porcentaje de salinidad en agua recogidas.

Obviamente, la metodología empleada no es la más efectiva, teniendo en cuenta que las mediciones dependen en gran parte del rango definido. Es decir, los valores obtenidos no son absolutos, sino relativos al marco de referencia preestablecido. Además, existen otras soluciones como los ácidos que también son electrolitos. No obstante, para el objetivo de la práctica, a modo de prototipo, es la implementación más sencilla y barata.




Véase una tabla con las mediciones realizadas durante la calibración del sensor. En este caso, el sensor devuelve un valor de salinidad de ~10 - 15%.

**SENSOR DE LUZ**

El sensor de luz funciona con un fotodiodo, que según recibe luz transmite una corriente u otra. Para la calibración de este sensor se ha medido el voltaje en diferentes condiciones de iluminación y según los valores medidos, se han establecido rangos de luminosidad para que dependiendo de los valores que lea el sensor, devuelve el estado de luz correspondiente.

| | Situación 1 | Situación 2 | Situación 3 | Situación 4 |
|---|---|---|---|---|
| **Condiciones de iluminación** | Sombra | Sol (poco de sombra) | Zona acristalada | Sol |
| **Vo** | 122mV | 360mV | 1.37V | 3.65V |

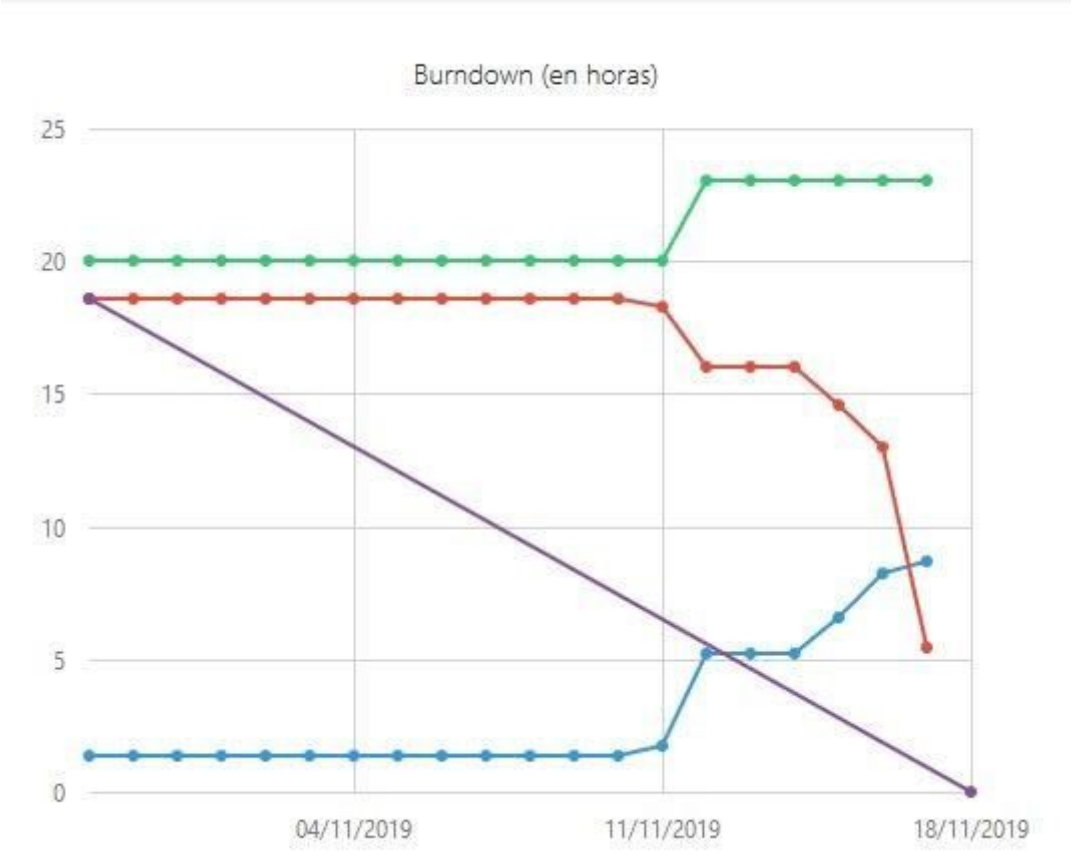
## CDIO

**ACELERÓMETRO**

En el caso del acelerómetro probaremos varios valores para la dirección **0x1F** en el cual se programa la sensibilidad del mismo a los cambios de aceleración. Se pueden situar valores entre 1 y 255. En nuestro caso usaremos 0x02 para detectar cambios leves en la aceleración.

Al tratarse de un sistema de seguridad, es mejor captar movimientos leves. Pero el valor 0x01 se activaba incluso cuando no era necesario. Con el valor 0x02 el funcionamiento es óptimo.

# DIAGRAMA DE BURNDOWN




Siendo la línea roja el trabajo restante, vemos que en este sprint hemos sobreestimado la cantidad de trabajo.

Véase el archivo original en el directorio *"Sprint3/img/Burndown.png"* en el repositorio o en la sección de documentos de la aplicación de gestión de proyectos ágiles Worki. Para un seguimiento más detallado, acceder a la sección de *Seguimiento*, subsección *Dashboard* en dicha aplicación.

# DAILY SCRUM

Véase el documento adjunto en el directorio *"Sprint3/Actas_Daily_Scrum.pdf".*