



**Universidad Nacional de La Matanza
Departamento de Ingeniería e Investigaciones Tecnológicas**

Sistemas Operativos Avanzados

SISTEMAS EMBEBIDOS-IoT

Días de Cursada: Lunes

Turno: Noche

Docentes: Lic. Graciela de Luca
Ing. Waldo Valiente
Ing. Esteban Carnuccio
Ing. Mariano Volker
Ing. Sebastian Barillaro

Integrantes:
Luciano, Esteban DNI 35.537.034
Terraza, Rodrigo DNI 38.690.721
Viqueira, Martin DNI 35.429.075

Trabajo Práctico: Sistemas Embebidos y Android – IoT

Objetivo: Desarrollar un sistema que mediante la medición de la temperatura, humedad e iluminación que se tenga en el ambiente, automatice el secado mediante un calentador y un ventilador de una prenda del usuario de forma óptima.

Descripción del funcionamiento - Arduino:

El funcionamiento se basa en medir distintas magnitudes climáticas para así determinar si debe iniciar el secado. Se mide la temperatura, la humedad y la iluminación del ambiente. Si la humedad de la prenda se encuentra en un valor alto significa que el proceso de secado todavía no se terminó por lo que se activa el calentador y el ventilador. La temperatura se mide para determinar la velocidad del ventilador, ya que una temperatura elevada ayuda al proceso de secado y no es necesario depender tanto del ventilador. La iluminación se utiliza para determinar si se encuentra en una posición donde se pueda aprovechar el sol y consumir menos energía apagando el ventilador o calentador. Se tiene un led, el cual cuando se determina que el proceso está terminado, se enciende para avisar al usuario que se terminó el proceso. Por último se tiene una pantalla en la cual el usuario puede ir observando los valores que se están sensando en el momento para que tenga una idea estimada de cuando se terminara el proceso.

Descripción del funcionamiento - Android:

El funcionamiento de la APK es bastante sencillo, utilizando de base una APK especializada en la búsqueda y emparejamiento de dispositivos mediante Bluetooth.

Consta de 3 pantallas principales, y un Servicio:

- **MainActivity:** Activity principal, el cual detecta sensores de Android y permite el acceso a los otros Activities.
- **ComunicacionService:** Servicio creado para la conexión automática por Bluetooth, y para el envío y recepción de información desde Arduino.
- **DatosActivity:** Activity para mostrar la información de los sensores y actuadores del Arduino, junto con un cálculo estimado del tiempo de finalización del secado. Necesita de presionar el botón “Cargar Datos”, para llamar al Service para que obtenga los datos.
- **ConfigActivity:** Activity para realizar configuraciones de los umbrales de Fin de los sensores de Arduino, y para prender y apagar distintos Actuadores. Necesita de presionar el botón “Cargar Datos”, para llamar al Service para que obtenga los datos; y también se agrega el botón “Restaurar Valores”, para decirle a Arduino que ponga sus valores por defecto.

En esta APK, se realizan mediciones con los siguientes sensores (los cuales fueron elegidos por estar en cualquier Android), con su correspondiente reacción en Arduino:

- **Acelerómetro:** Se utiliza el conocido “Shake”, el que Arduino captura y muestra por la pantalla LCD la leyenda “Shake reconocido”.
- **Proximidad:** Cuando el usuario se acerca lo necesario a dicho sensor, este manda un mensaje al Arduino, el cual prenderá o apagará el calentador a voluntad.
- **Luminosidad:** Cuando el dispositivo Android detecta que no hay suficiente luz en el ambiente, envía un mensaje al Arduino para prender su LED verde.

Pantallas Android:



Diagrama de Componentes

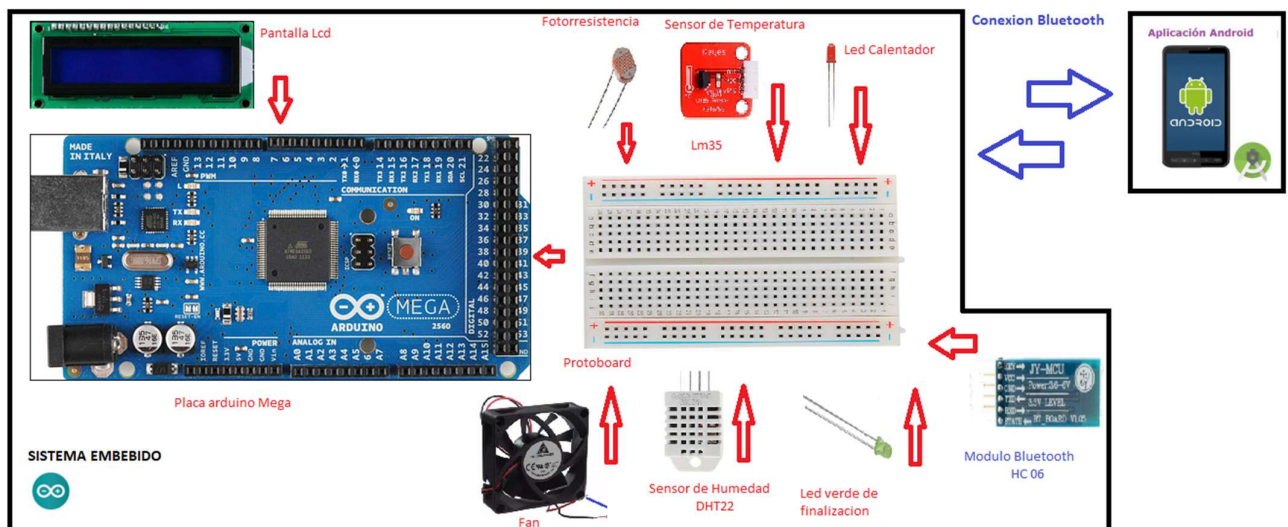
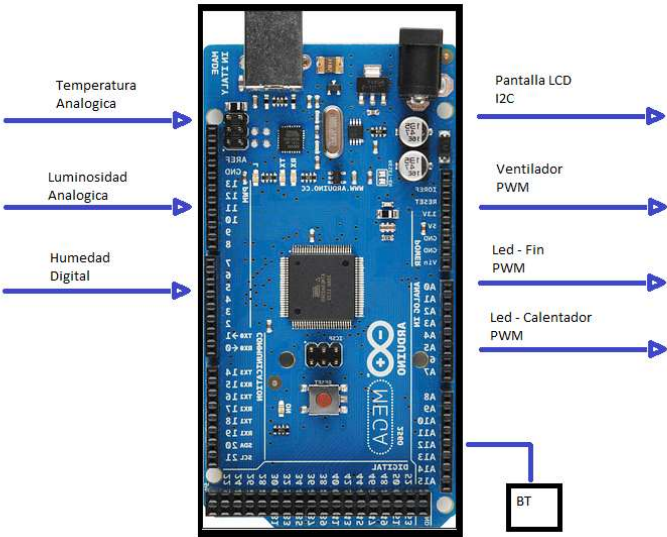
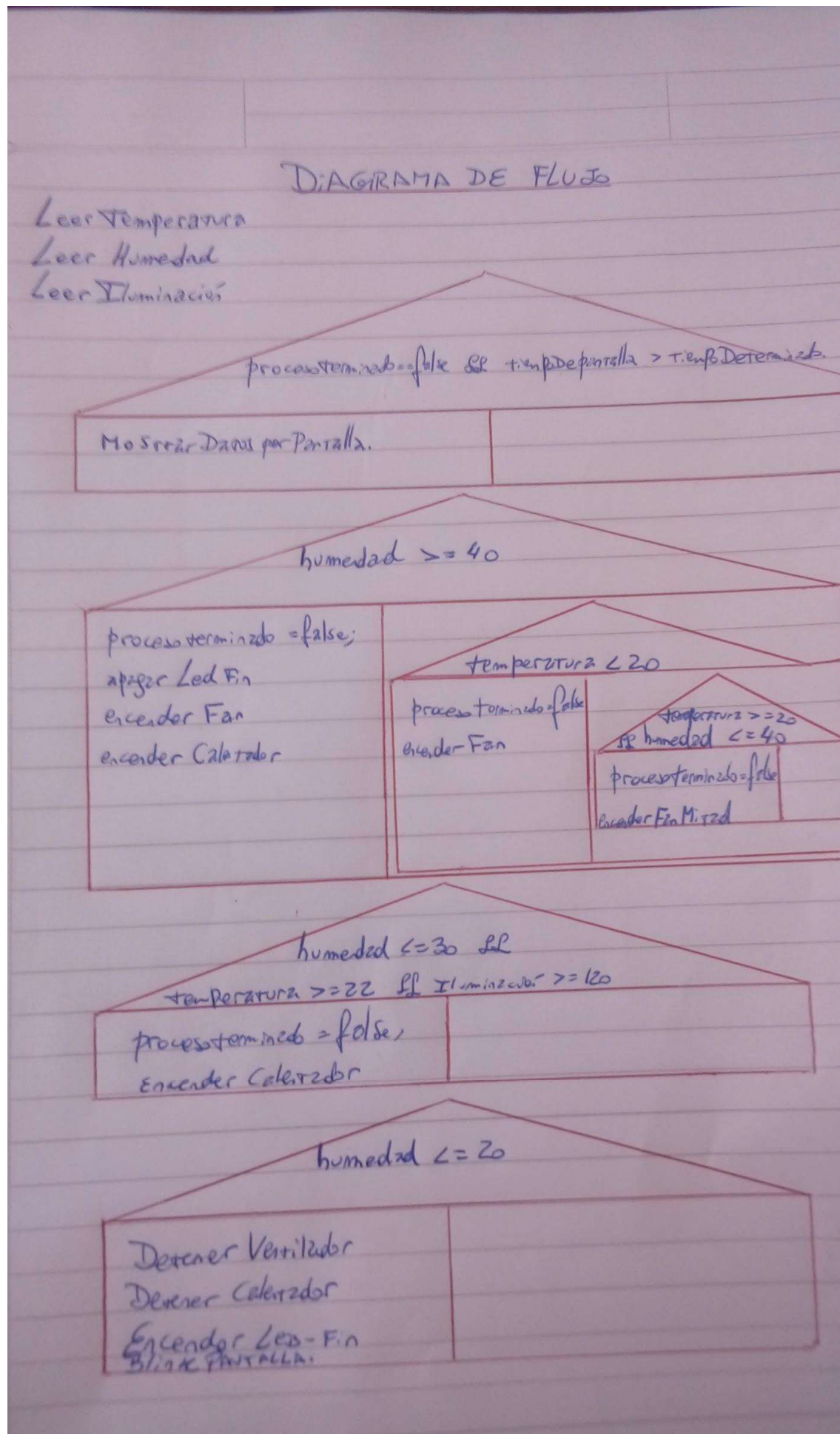
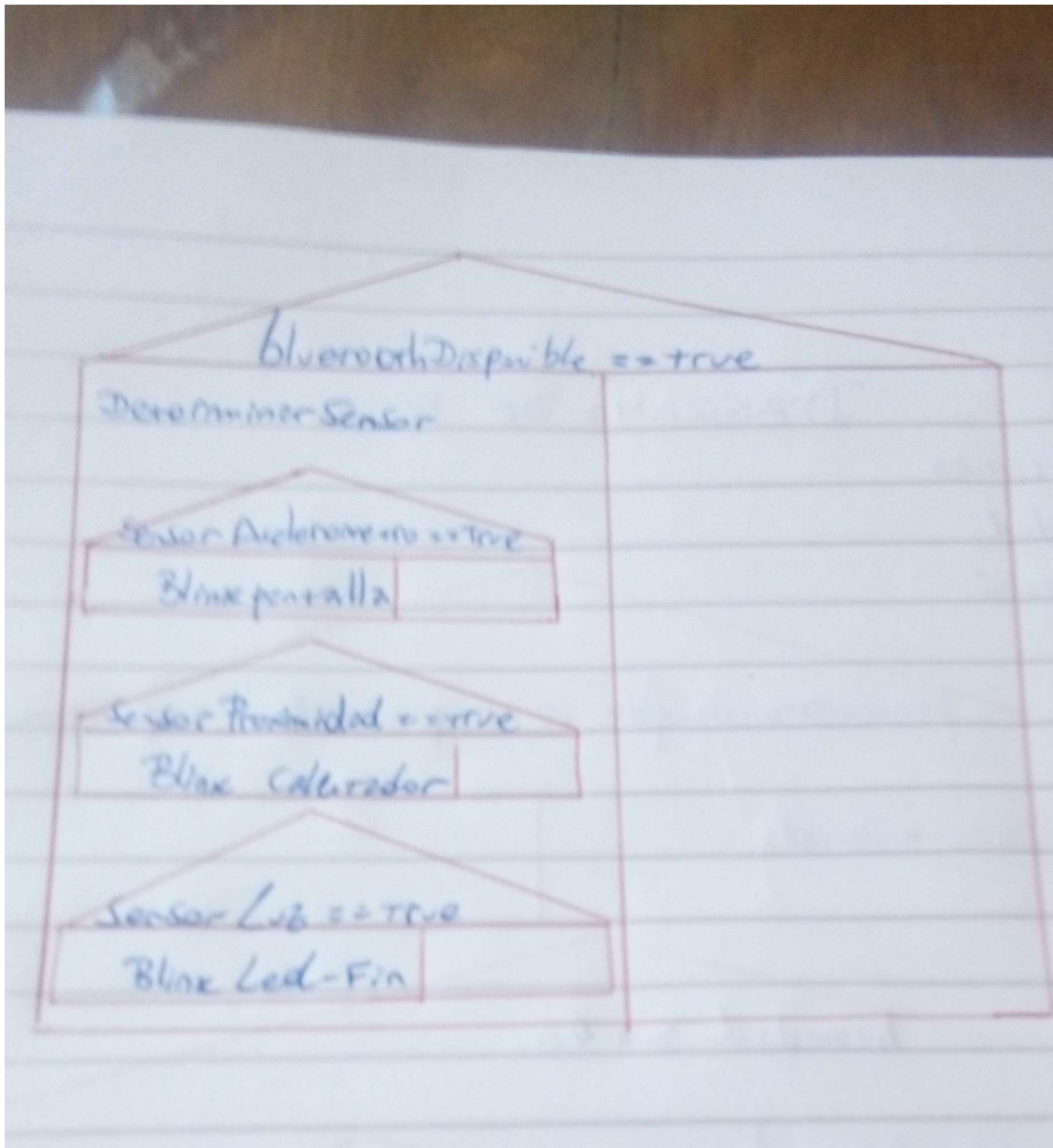
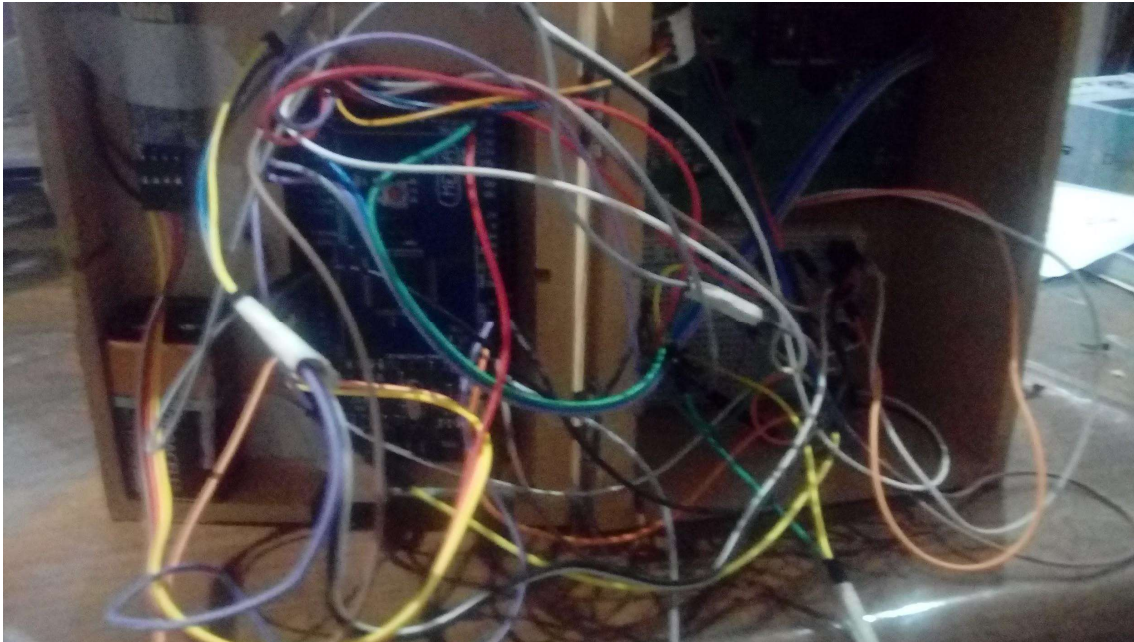


Diagrama de Señales









Hardware IoT

Pantalla LCD - I²C

El I²C es un bus maestro-esclavo. La transferencia de datos es siempre inicializada por un maestro; el esclavo reacciona. El I²C precisa de dos líneas de señal: reloj (CLK, Serial Clock) y la línea de datos (SDA, Serial Data). La pantalla nuestra cuenta con 2 pines más - Vcc y Gnd. Vcc es de 5 volts y está conectado a unos de los pines de la placa. Gnd también está conectado directamente a uno de los pines de la placa.

I²C utiliza comunicación del tipo half-duplex por lo que no se puede enviar y recibir al mismo tiempo, por eso usa solo un pin de data (SDA).

Modelo de la placa = LCM1602.

La placa viene con un módulo de expansión de E/S de 8 bits que es PCF8574.

Librerías utilizadas:

- Wire.h
- LiquidCrystal_I2C.h

Métodos utilizados:

- ```
addr, en, rw, rs, d4, d5, d6, d7, bl, blpol
```
- LiquidCrystal\_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
  - lcd.begin(columnas, filas);
  - lcd.clear();
  - lcd.setCursor(columna, fila);
  - lcd.print("Mensaje");
  - lcd.backlight();

Addr = Es la dirección del módulo de expansión E/S del LCD. En este caso es 0x27.

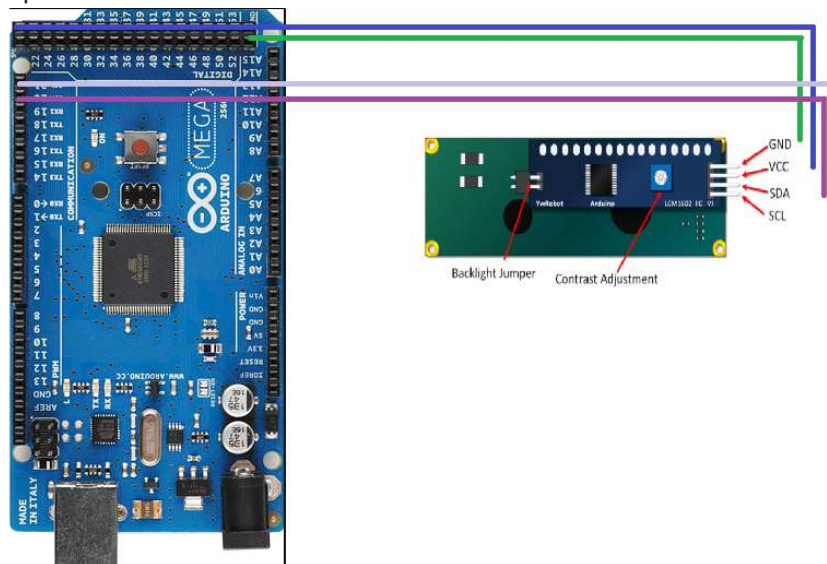
En = Enable. Es para activar el uso del módulo.

Rw = Read/Write. Es el pin conectado al módulo para realizar la lectura-escritura.

D4,D5,D6,D7 = Son los pines 0,1,2,3 para hacer el mapeo del LCD en el módulo.

BI = Backlight. Es para el control de la luz trasera de la placa.

BI pol = Si está activado ese control.

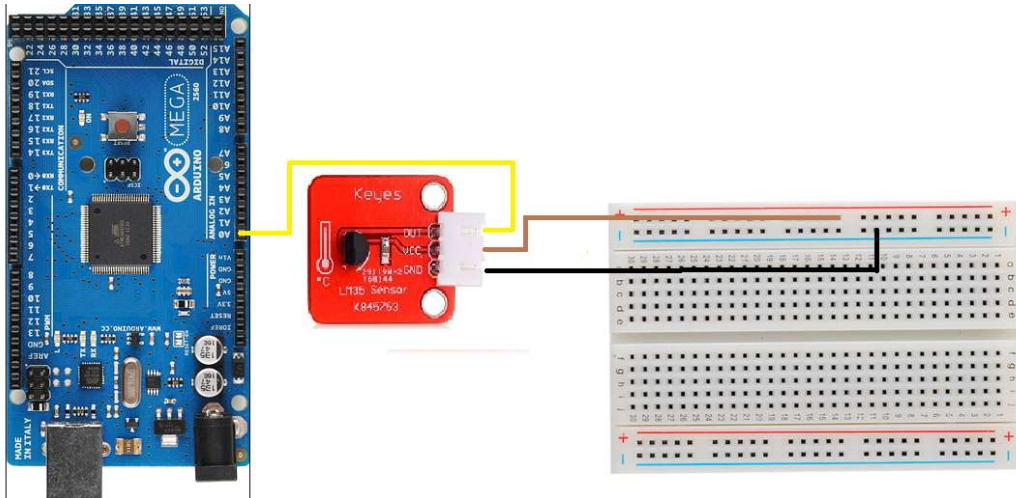




## Sensor de Temperatura - Analógico

El **LM35** es un sensor de temperatura con una precisión calibrada de 1 °C. Su rango de medición abarca desde -55 °C hasta 150 °C. La salida es lineal y cada grado Celsius equivale a 10 mV, por lo que se debe hacer una conversión para tener el valor que se está registrando. Al utilizar `AnalogRead`(pin del Lm35), se obtiene un valor entre 0 y 1023, es decir 1024 valores. Cómo utiliza 5 volts de alimentación, se divide el voltaje en la cantidad de valores posibles.  $5/1024$  genera una precisión de 0,0048 volts. Como 1°C equivale a 10mV, la ecuación final termina siendo:  $(10\text{mV} * 5 * \text{analogRead}) / 1024$ .

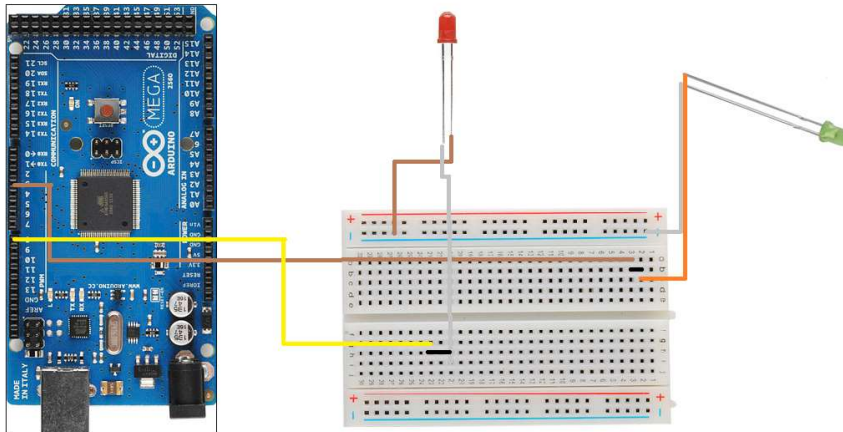
No utiliza librerías ni métodos especiales, ya que es una lectura analógica de un pin.



### Leds - PWM

Hay dos Leds, uno verde, el cual representa el finalizado del proceso y uno rojo, el cual representa el heater. Se debe utilizar una resistencia para conectarlos ya que si la tensión es inferior a un valor determinado (Valor de polarización directa), el Led no se enciende y por el contrario si es superior a este valor el Led se rompe. El led es un diodo, por lo que tiene polaridad, entonces se debe conectar de una manera específica. Es posible conectarlo directamente a la placa ya que las salidas del Arduino es de 20mA.

Para iluminar los Leds se utiliza PWM (Modulación por ancho de pulso), que es una técnica en la que se modifica el duty cycle de la señal para controlar la energía que se envía.



### Sensor de Humedad - Digital

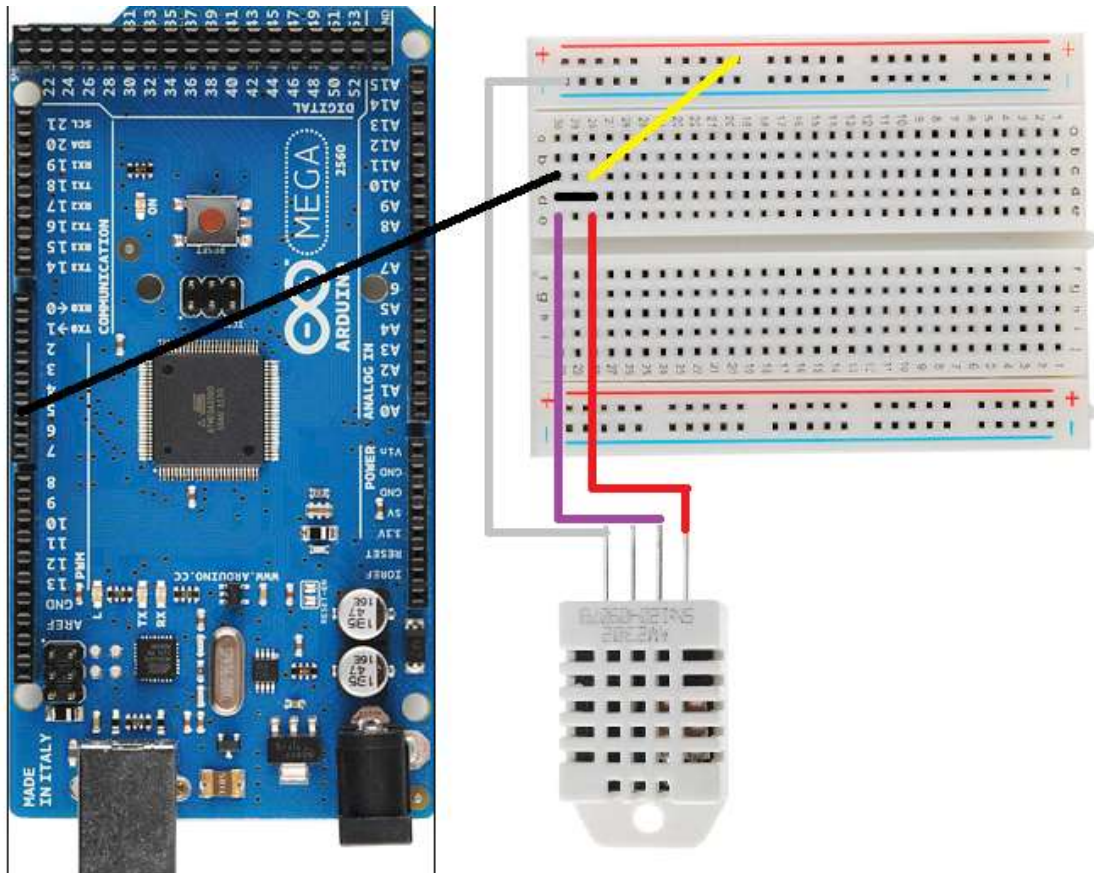
El DHT22 es un sensor que permite medir temperatura y humedad de forma digital. La temperatura posible que mide es de  $-40^{\circ}\text{C}$  a  $125^{\circ}\text{C}$ , la humedad entre 0 y 100 %, con una precisión de entre 2 y 5%. Genera 2 muestras por segundo. El mismo cuenta con 4 pines. En este caso se utilizan 3 - Vcc, Gnd y el Output. Para utilizarlo se debe poner una resistencia de 10k entre el Vcc y el pin de Output, para que los valores que se generan queden entre valores aceptables.

Librería utilizada:

- DHT.h

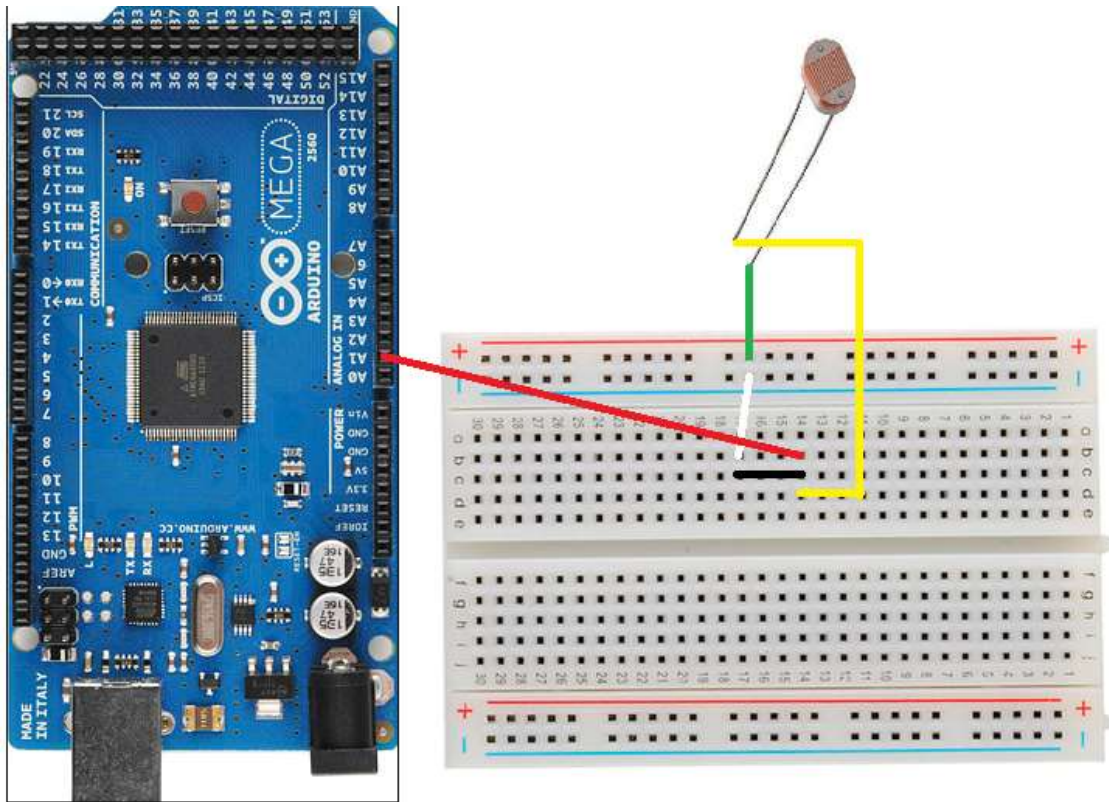
Metodos utilizados:

- DHT sensorDeHumedad(pinDHT, DHTTYPE); Inicializar el sensor con el pin y el tipo(Dht22)
- sensorDeHumedad.begin();
- sensorDeHumedad.readHumidity()



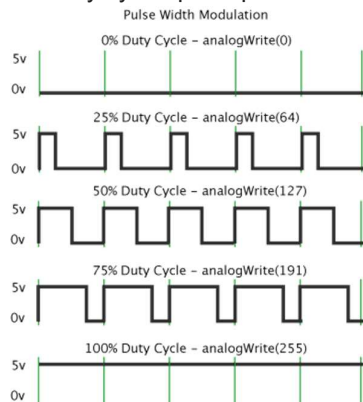
## Sensor de Luz - Analógico

Se utiliza una fotoresistencia. Es un componente cuya resistencia varía según la cantidad de luz que percibe. La relación entre la intensidad lumínica y el valor de la resistencia no es lineal. Cuanta más luz percibe hay menor resistencia eléctrica y cuanto menor es la luz mayor resistencia eléctrica. Es un sensor analógico por lo que para conocer su valor solo es necesario realizar un `AnalogRead()`;



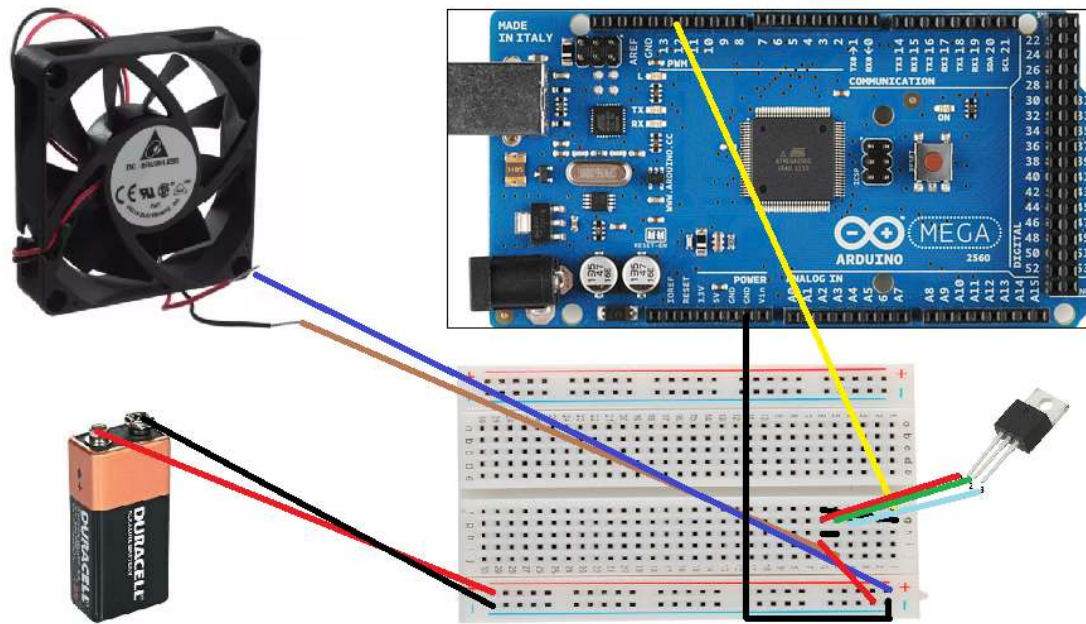
### Ventilador - PWM

Se utiliza Pwm para controlar la velocidad del ventilador. Con un AnalogWrite se indica el porcentaje del duty cycle que representa las distintas velocidades que puede tomar.



Se utiliza además un transistor (Tip 122) que funciona como una compuerta entre el arduino y la batería externa que se utiliza para darle potencia al Fan ya que se necesitan 9 volts para mover el Fan de forma adecuada y el arduino tiene 5 volts. El arduino puede indicar la potencia necesaria sin tener que estar directamente conectado. El mismo tiene 3 pines - base, colector y emisor. El base se conecta con el arduino mediante una resistencia de 1K para protegerlo. Luego entre el colector y el emisor se introduce un diodo para permitir que la corriente sólo circule en una dirección. El emisor se conecta con el Gnd y el colector se conecta con el pin tierra del Fan. El otro pin del fan, el de corriente, se conecta a la protoboard en la parte positiva la cual tiene una batería de 9v entregando potencia.







### **Conclusiones y Experiencias**

A lo largo del TP hemos tenido varios contratiempos y problemas que hemos ido solucionando, los cuales detallo algunos como ejemplo:

- En Arduino:
  - Usamos varios sensores y actuadores que necesitaban corriente de 5v proveniente desde la placa Arduino Mega, pero no teníamos los puertos suficientes, por lo que decidimos realizar un puente entre uno de los laterales de la Protoboard y el puerto de 5v del Arduino, para conectar en esa hilera lateral los distintos sensores y actuadores en exceso que tenemos.
  - El proyecto debe poseer un calentador que funcione a la par con el Ventilador, para acelerar el secado, pero no hemos podido hacerlo funcionar por falta de recursos. Quisimos armar una hilera de resistencias que recibiendo corriente desde la batería de 9v (que también le da energía al ventilador mediante las líneas laterales de la Protoboard), pero los mismos no quisieron funcionar. Hemos decidido tomar la propuesta de un profesor de reemplazar dicho calentador con LED que simulen su funcionamiento, aunque esperamos en un futuro hacerlo funcionar tal como fue pensado.
  - Para complementar con la falta del calentador en la presentación, se agregó una pantalla LCD que muestre por intervalos los valores que reciben los sensores, como también un texto para cuando el proceso se lo determine cómo “Finalizado”.
- En Android:
  - Un problema que se ha presentado es en el envío de mensajes mediante Bluetooth al receptor HC06 del Arduino, ya que notamos que dicho Sensor acumula los mensajes en su buffer previo a que Arduino los consuma, haciendo poco confiables las interacciones, puesto que hay posibilidades de que los sensores de Android envíen varias señales (por ejemplo, varios cambios en la luz de ambiente), los cuales terminan siendo “descartados”. Para solucionar esto, se ha cambiado la forma en que Arduino interpreta dichos mensajes (en vez de una comparación literal, realizamos un “startsWith”), cumpliendo el objetivo.

Como conclusión, podemos decir que fue un proyecto desafiante e interesante, por el uso nuevas tecnologías actuales, lo que nos aporta experiencia en este rubro y nos demuestra las infinitas posibilidades que tienen los proyectos IoT.