

EVALUACIÓN DE APRENDIZAJE 3

Nombre y apellido: Tomás Sánchez Grigioni.

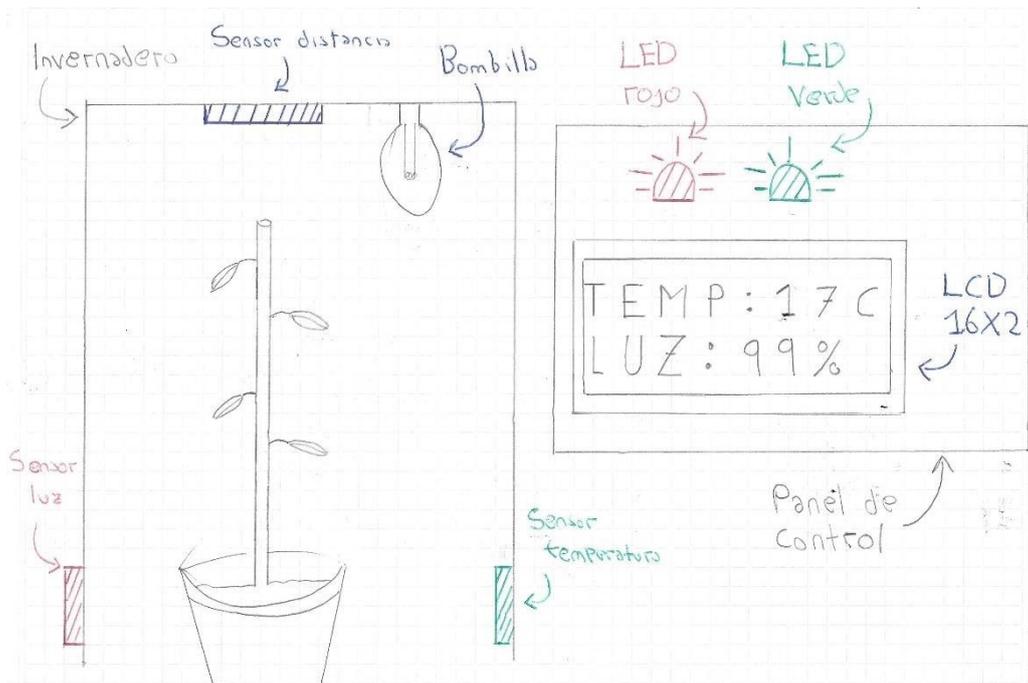
Curso: martes – noche.

1 INTRODUCCIÓN	2
1.1 DESCRIPCIÓN FUNCIONAL	2
1.2 POSIBLES MEJORAS	4
2 DIAGRAMA DE ESTADOS	5
3 SENSORES Y ACTUADORES	6
3.1 SENSOR DE TEMPERATURA TMP36	6
3.2 FOTORRESISTENCIA LDR (SENSOR)	7
3.3 SENSOR DE DISTANCIA POR ULTRASONIDO	8
3.4 BOMBILLA DE LUZ (ACTUADOR)	10
3.5 LED (ACTUADOR)	11
3.6 LCD DE 16X2 (ACTUADOR)	12
4 DIAGRAMA DE CONEXIONES	14
5 FUNCIONAMIENTO SIMULACIÓN	14

1 INTRODUCCIÓN

1.1 DESCRIPCIÓN FUNCIONAL

El sistema embebido tiene como objetivo el control de un invernadero autónomo. Primero que todo, el sistema se diseñó para los requerimientos de una planta genérica, por lo que si se desea implementar este sistema en el mundo real se debe evaluar los requerimientos de una planta específica y adaptar los valores críticos. Para explicar su funcionamiento primero se presenta un diagrama de planta donde se indican las distintas partes que componen al invernadero.



El sistema consta de tres sensores distintos:

- Sensor de temperatura.
- Sensor de luz.
- Sensor de distancia.

Además, se utilizan tres actuadores:

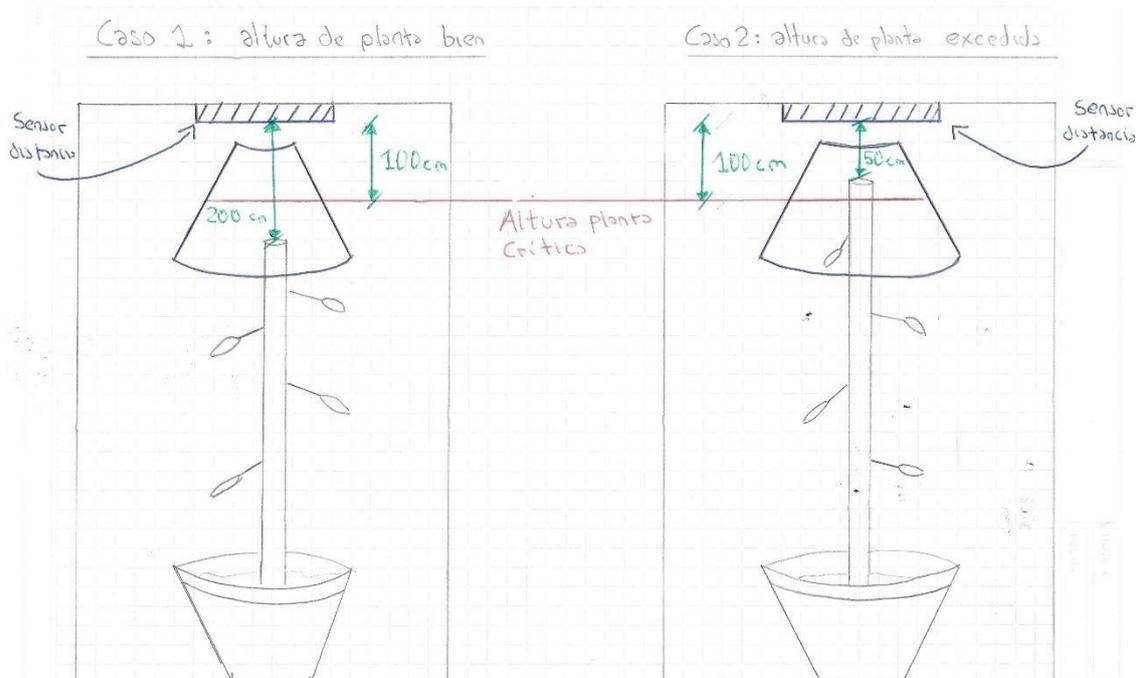
- Bombilla.
- LEDs, una de color rojo y otra de color verde.
- LCD de 16X2.

Para la implementación se hizo uso de un temporizador por hardware, para que cada 250 ms se genere una interrupción y así se realiza el senseo, evaluación de los valores obtenidos y una posterior activación de los actuadores de acuerdo con el caso. El valor de 250 ms se eligió con propósito demostrativos del proyecto, si se tuviera que implementar en la vida real se hubiera optados por valores muchos más altos, de varios minutos. Esto es porque los valores de temperatura y de luz tienen cambios significativos en intervalos de tiempos mayores a 250 ms, y de esta forma evitar estar sensando repetidamente el mismo valor innecesariamente.

La forma en que interactúan los sensores con los actuadores se describe a continuación:

- El sensor de luz mide el nivel de luz del ambiente, luego se evalúa si el nivel de luz se encuentra por encima de un valor crítico. En el caso de que sea cierto, la bombilla de luz debe estar apagada y en caso contrario se enciende. Esto es para que la planta reciba constantemente como mínimo un determinado nivel de luz.
- El sensor de temperatura mide la temperatura del ambiente, luego evalúa si la temperatura se encuentra dentro de la región de temperaturas en que la planta puede crecer (delimitado por dos valores críticos: un límite inferior y un límite superior). Si el valor sentido se encuentra dentro de la región entonces el LED rojo se apaga, y en caso contrario se enciende para indicar que se deben tomar medidas.

- El sensor de distancia mide la distancia a la planta, de esta forma se puede conocer la altura de está realizando algún cálculo. Se evalúa si la planta se encuentra por debajo de un cierto nivel crítico, si es así se apaga el LED verde y en caso contrario se enciende. Se incluye un diagrama de planta con un caso en que la planta excede este valor crítico y otro en el que no.



Con este sensor se puede registrar el avance de la altura de la planta en el tiempo. También, si existe alguna relación entre la madurez de la planta y su altura, determinar el estado de la planta.

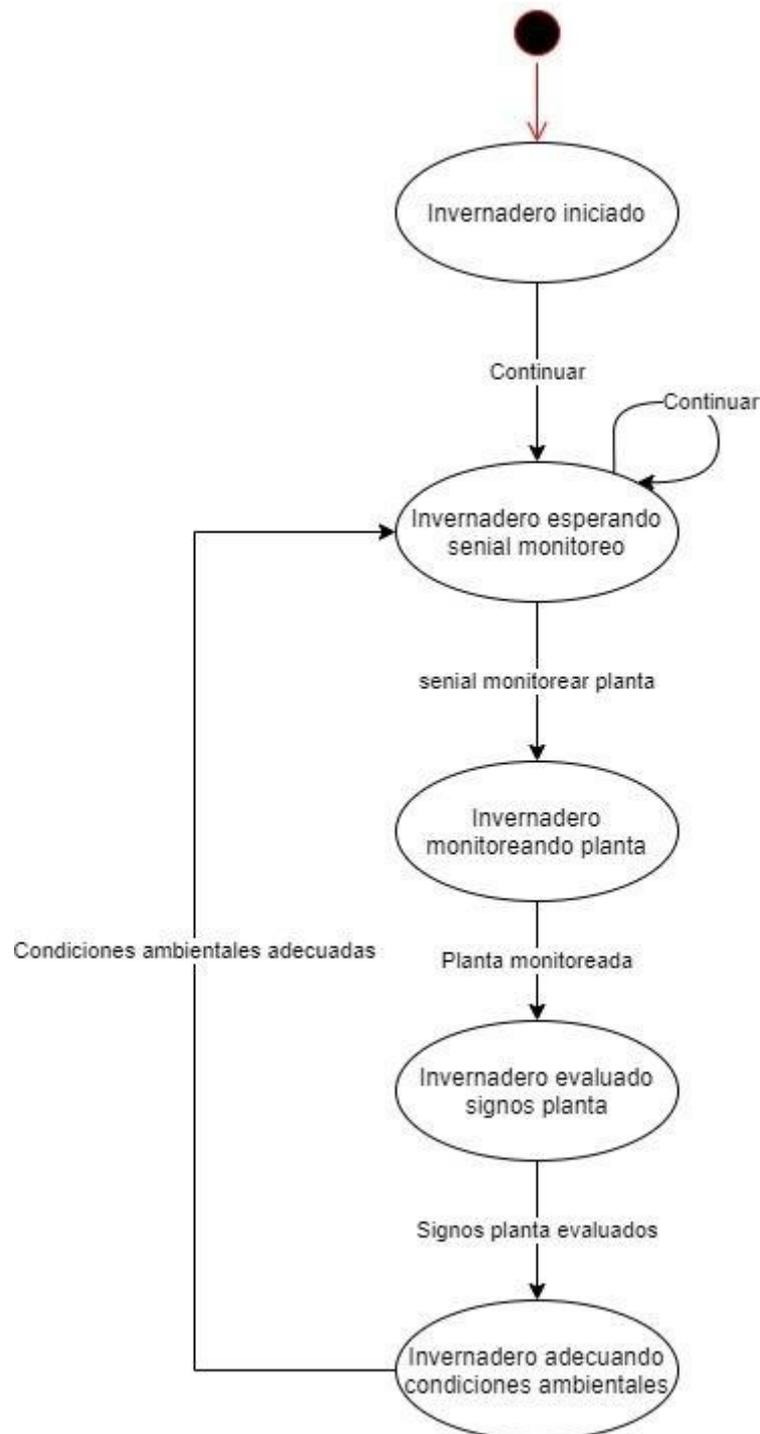
- En cada ronda de senseo se actualiza el LCD con los valores de temperatura y nivel de luz, expresado en grados centígrados y porcentaje correspondientemente.

1.2 POSIBLES MEJORAS

Como ya se mencionó, el diseño se realizó de acuerdo con los requerimientos de una planta genérica. Por esto, la primera mejora al sistema es adaptar los valores de acuerdo con los requerimientos de una planta específica. Además, se desea incluir algún actuador que logre regular la temperatura del invernadero, así no se tiene que depender de que una persona realice alguna acción para nivelar la temperatura. Por último, incluir un módulo para conectarse con un servidor y un dispositivo móvil. El servidor se utilizaría para registrar los valores medidos, para así poder analizarlos y poder generar informes. El móvil es para lograr monitorear el estado del invernadero a la distancia, y que no sea necesario tener que consultarlo directamente en el panel de control.

2 DIAGRAMA DE ESTADOS

Se presenta un diagrama de estados de la máquina de estados implementada



- **Invernadero iniciado:** primer estado en el que entra el sistema embebido. Aquí solo se realizan las inicializaciones. En caso de error, se puede retornar a este estado para reiniciar el sistema.
- **Invernadero esperando senial monitoreo:** equivale al estado ocioso, solo espera a que se produzca una interrupción cada 250 ms.

- **Invernadero monitoreando planta:** se realizan las lecturas de los sensores.
- **Invernadero evaluando signos planta:** se realizan las comparaciones de los valores leídos con las condiciones predefinidas.
- **Invernadero adecuando condiciones ambientales:** según los resultados del estado anterior se actualizan o no los actuadores.

3 SENSORES Y ACTUADORES

3.1 SENSOR DE TEMPERATURA TMP36

- **Tipo sensor:** Analógico
- **Forma y magnitud de trabajo:** Este sensor devuelve un valor de tensión proporcional a la temperatura a la que es sometido. Primero que todo, el sensor al estar conectado a un pin analógico devuelve un valor entre 0 y 1023, siendo 1023 un valor de 5V y 0 un valor de 0V. Entonces para conocer el valor de tensión leído (por regla de tres simple):

$$V_{entrada} = \frac{5}{1023} * lectura V$$

Por las especificaciones técnicas conocemos que por cada grado de temperatura la salida de tensión es de 10mV. De esta regla deducimos que una tensión de 1V equivale a 100 °C. Conociendo el valor de tensión leída, podemos deducir la temperatura exterior.

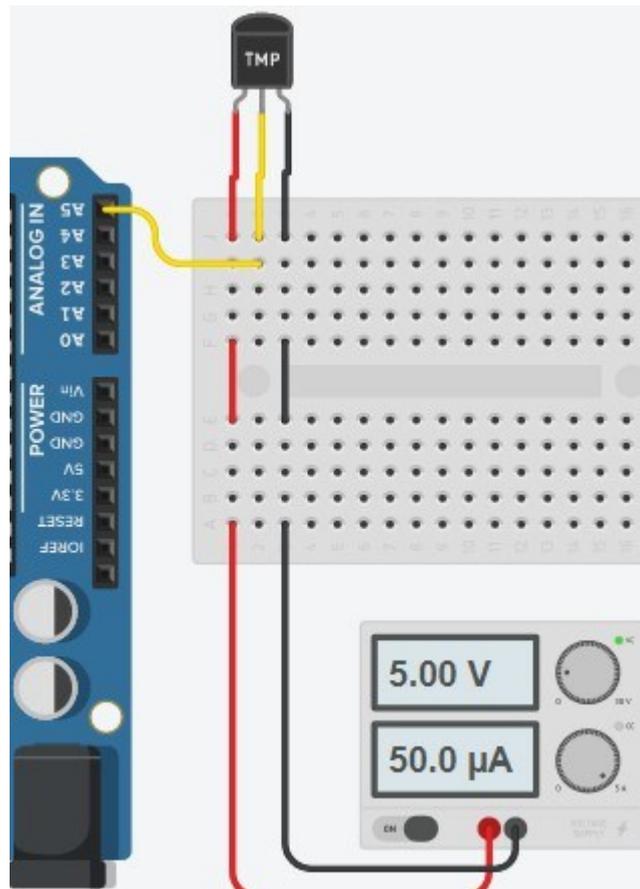
$$Temperatura = V_{entrada} * 100 \text{ } ^\circ C$$

$$Temperatura = \frac{5}{1023} * lectura * 100 \text{ } ^\circ C$$

Sin embargo, el sensor mide temperaturas desde -50°C hasta 125°C por lo que hay que restar 50°C para que se concuerde una lectura de 0V con una temperatura de -50°C.

$$Temperatura = \frac{5}{1023} * lectura * 100 \text{ } ^\circ C - 50 \text{ } ^\circ C$$

- **Uso en el simulador:** Para utilizar dicho sensor en el simulador es necesario conectar la pata derecha a un borne negativo de una fuente de alimentación de 5V y la pata izquierda al borne positivo de este. Por último, la pata intermedia se conecta a una entrada analógica del Arduino UNO.

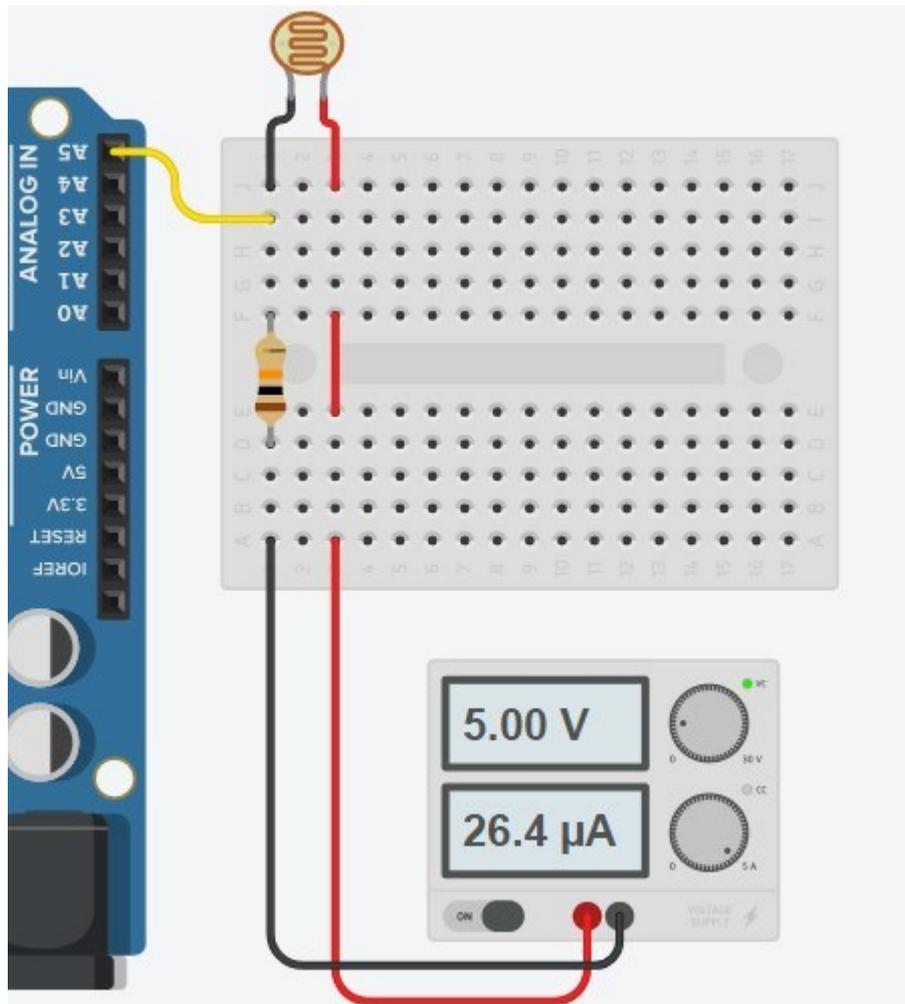


Para el código se utiliza la fórmula explicada en la sección anterior

```
float leer_temperatura(int sensor)
{
    return 0.48828125 * analogRead(sensor) - 50;
}
```

3.2 FOTORRESISTENCIA LDR (SENSOR)

- **Tipo sensor:** Analógico
- **Forma y magnitud de trabajo:** Es un componente cuya resistencia varía sensiblemente con la cantidad de luz percibida. La relación entre la intensidad lumínica y el valor de resistencia no es lineal, pero se sabe que son inversamente proporcional (mayor luz conlleva menor resistencia eléctrica y menor luz conlleva mayor resistencia eléctrica). Su funcionamiento se basa en el semiconductor Sulfato de Sodio, que cuando la luz incide en el dispositivo provoca que los electrones adquieran suficiente energía para saltar a la banda de conducción y así reducir la resistencia por la conducción de electricidad.
- **Uso en el simulador:** Para utilizar dicho sensor en el simulador es necesario conectar la pata derecha al borne positivo de una fuente de alimentación de 5V y la pata izquierda al borne negativo de la misma fuente, pero con una resistencia de 10KΩ intermedia. A su vez, se conecta la pata izquierda a una entrada analógica del Arduino UNO.



El valor que devuelve el sensor depende de la resistencia que se utilice. Se demostró empíricamente que con dicha resistencia se consiguen valores que varían entre 54 y 974, por lo que cuando se lee se realiza un mapeo para expresarlo como un porcentaje.

```
int leer_porcentaje_luz(int sensor)
{
    // Realizamos un mapeo para tener el porcentaje de luz recibido
    int valor_luz = analogRead(sensor);
    return map(valor_luz, VALOR_MIN_SENSOR_LUZ, VALOR_MAX_SENSOR_LUZ, 0, 100);
}
```

3.3 SENSOR DE DISTANCIA POR ULTRASONIDO

- **Tipo sensor:** Digital
- **Forma y magnitud de trabajo:** Este sensor mide la distancia de un objeto en base a ondas ultrasónicas emitidas. Se compone por un transmisor y un receptor, el transmisor emite las ondas ultrasónicas que viajan por el espacio, que al encontrar un obstáculo son reflejadas y captadas por el receptor. En base al tiempo transcurrido se calcula la distancia del obstáculo en cm. Para calcular la distancia se utiliza

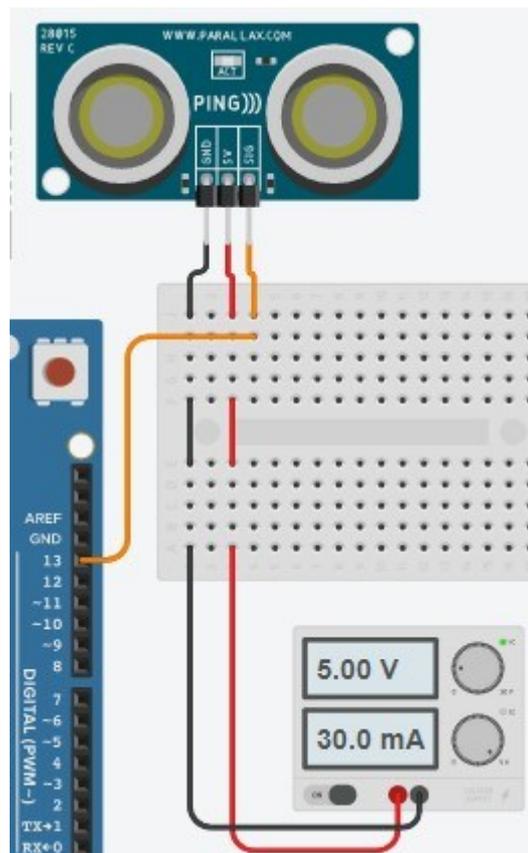
$$\text{Distancia} = \text{Velocidad} * \text{Tiempo}_{ida}$$

Sin embargo, el tiempo de ida no se conoce pero si el tiempo total. El tiempo de ida resulta ser la mitad del tiempo total. Además, la velocidad de la onda transmitida es la velocidad del sonido.

$$Distancia = \frac{0,03446 * Tiempo_{total}}{2} \text{ cm}$$

Para sensar se debe enviar una señal de un ancho de pulso de 10 μ s a la pata de trigger para iniciar la transmisión de la onda. Al finalizar la emisión de la onda ultrasónica, por la pata de echo se inicia la transmisión de una señal. Una vez que el receptor detecte la onda interrumpe la señal de echo y la lleva a un estado bajo. En base al ancho del pulso resultante (tiempo de señal que estuvo en estado alto) se calcula la distancia. Si el ancho de pulso resultantes es mayor a 38ms, significa que el obstáculo se encuentra fuera de rango del sensor, devolviendo la distancia máxima que puede medir el sensor. En caso contrario, se calcula la distancia real del objeto.

- **Uso en el simulador:** Para utilizar dicho sensor en el simulador es necesario conectar la pata extrema izquierda a un borne negativo de una fuente de alimentación de 5V y la pata intermedia al borne positivo. Por último, se conecta la pata extrema derecha a una entrada digital del Arduino UNO. Notar que en el modelo de sensor usado se utiliza la señal echo y trigger por la misma pata.



El código de lectura es un poco más complejo porque se debe enviar pulsos con determinado ancho de pulso, para ello se hace uso de la función delayMicroseconds()

```
long leer_distancia(int trigger, int echo)
{
    pinMode(trigger, OUTPUT);

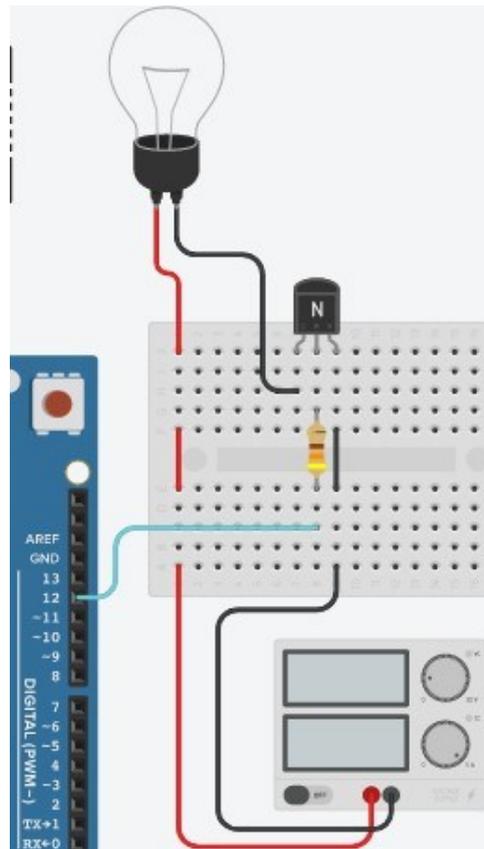
    // Reseteamos el trigger
    digitalWrite(trigger, LOW);

    // Mandamos la señal trigger
    digitalWrite(trigger, HIGH);
    delayMicroseconds(ANCHO_PULSO_TRIGGER_MICRO);
    digitalWrite(trigger, LOW);

    // Leemos la distancia a la que se encuentra el objeto
    pinMode(echo, INPUT);
    return 0.01723 * pulseIn(echo, HIGH);
}
```

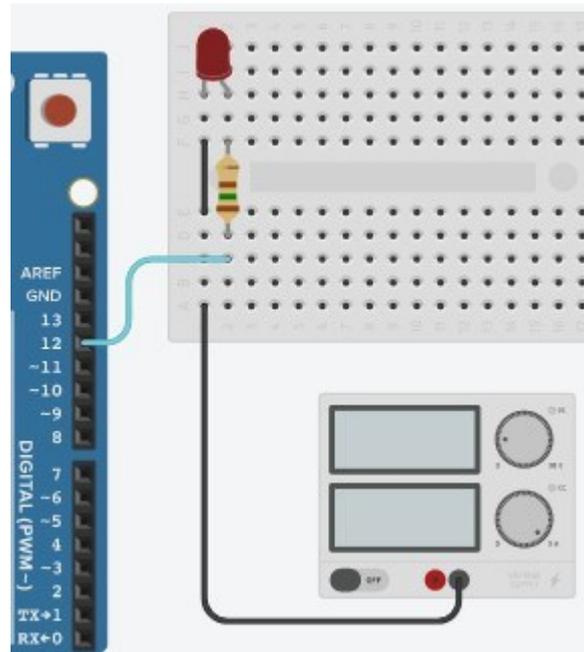
3.4 BOMBILLA DE LUZ (ACTUADOR)

- **Tipo actuador:** Digital
- **Forma y magnitud de trabajo:** Este actuador no es más que un conductor conectado a un metal especial, el cuál al calentarse irradia luz. De esta forma, al introducirse una corriente eléctrica provoca que se caliente el metal e ilumina.
- **Uso en el simulador:** Para utilizar este actuador se utiliza un transistor. Se conecta el borne positivo de la bombilla al borne positivo de una fuente de alimentación. El borne negativo de la bombilla se conecta al colector del transistor y el borne negativo de la fuente al emisor del transistor. Por último, se conecta la base transistor a un pin digital del Arduino UNO (a través de una resistencia) de forma que cuando se quiere prender la bombilla se transmite un HIGH usando digitalWrite() por este pin. Esto provoca que se haga un cortocircuito entre el emisor y el receptor para que pueda circular la corriente. Se hace uso de un transistor para que, si la bombilla a usar funciona con 220v entonces se pueda separar el Arduino UNO de la bombilla y la fuente.



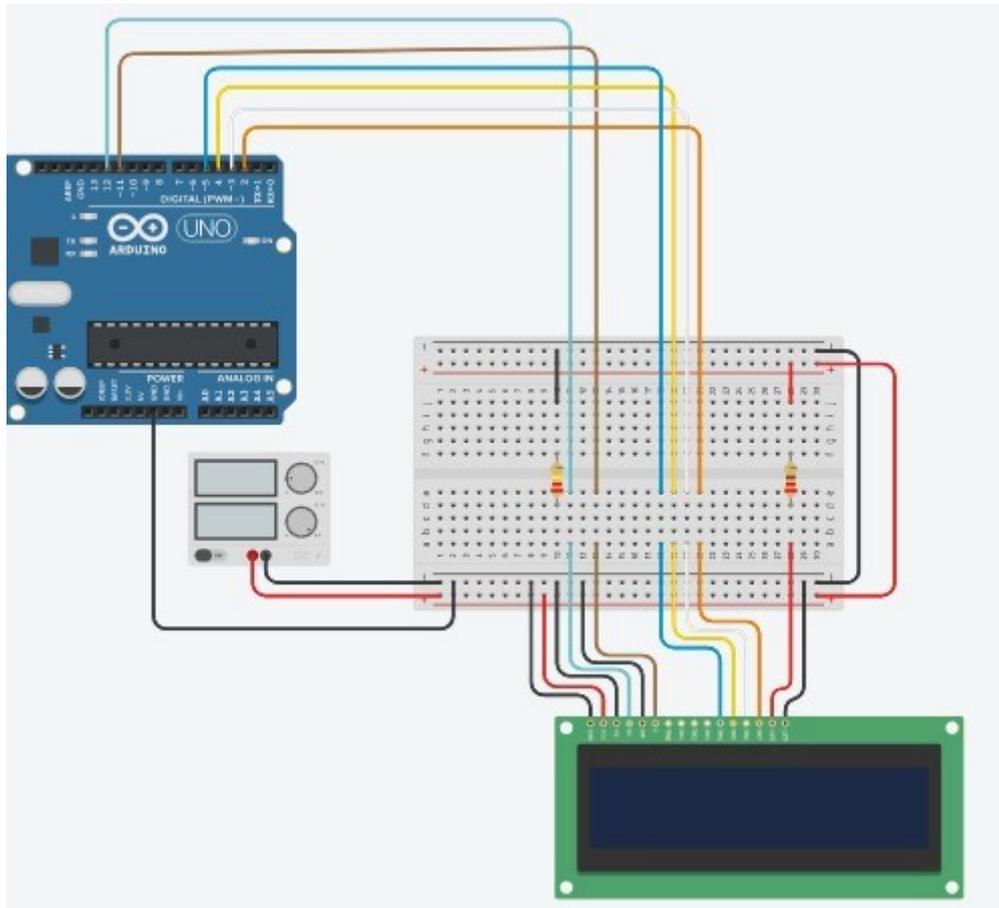
3.5 LED (ACTUADOR)

- **Tipo actuador:** Digital
- **Forma y magnitud de trabajo:** Un diodo es un componente que permite el paso de corriente en un solo sentido, desde el ánodo hacia el cátodo. De esta forma, si la corriente circula de forma contraria actúa como un interruptor cerrado. Un LED es un tipo especial de diodo que además emite luz por el paso de corriente de forma correcta.
- **Uso en el simulador:** Para utilizar este actuador se debe conectar el cátodo al borne negativo de una fuente de alimentación de 5V y el ánodo a un pin digital del Arduino UNO pero usando como intermediario una resistencia de 150 Ω para no quemar el LED por el exceso de corriente. Para encender y apagar el LED se hace uso de la función `digitalWrite()`.



3.6 LCD DE 16X2 (ACTUADOR)

- **Tipo actuador:** Digital
- **Forma y magnitud de trabajo:** Es un pequeño dispositivo con pantalla de cristal líquido que cuenta con dos filas, de dieciséis caracteres cada una, que se utiliza para presentar información alfanumérica. Cada carácter internamente se representa con una matriz de 8X5. A través, de señales y pines de datos se puede controlar el uso del LCD.
- **Uso en el simulador:** La conexión de este actuador es más compleja comparado con los vistos previamente. Por este motivo se incluye un diagrama de conexión para demostrar la conexión.

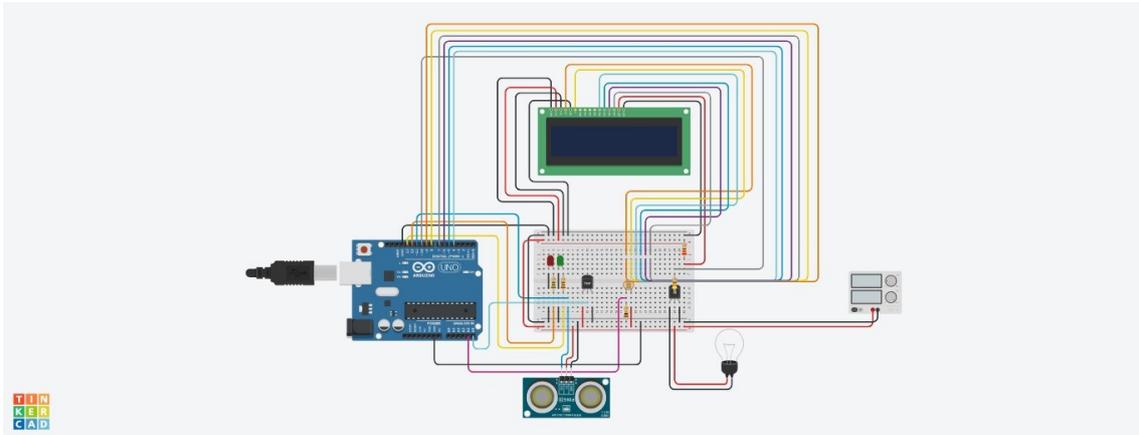


En cuanto a su utilización, se precisa de la biblioteca LiquidCrystal.h. Lo primero que se debe realizar es la iniciación del LCD con las funciones `lcd()` ingresando como parámetros los diferentes pines conectados y `begin()` con la cantidad de columnas y filas. Luego, principalmente se usan tres funciones. La primera es `clear()` para limpiar la pantalla. La segunda es `setCursor()` para poder elegir en que posición se desea imprimir por pantalla, indicando el número de fila y el número de columna. Por último, `print()` para imprimir un mensaje por pantalla en la posición indicada, indicando el mensaje.

En las fotos expuestas, como se hace uso de una fuente de alimentación externa se debe compartir la tierra de la fuente con el Arduino UNO para lograr controlar los sensores y actuadores. Además, para fines demostrativos se utiliza una mini placa de pruebas pero en el trabajo final se utiliza una placa de pruebas como se solicita.

4 DIAGRAMA DE CONEXIONES

El diagrama de conexiones se presenta a continuación



5 FUNCIONAMIENTO SIMULACIÓN

Durante la simulación se pueden realizar tres acciones:

1. Modificar el sensor de distancia. Si se pone el objeto a una distancia mayor a 100 cm el LED de color verde debe estar apagado y en caso contrario prendido.
2. Modificar el sensor de temperatura. Si la temperatura se la establece en un valor que se encuentra en el intervalo [5; 25] entonces el LED rojo debe estar apagado y en caso contrario debe estar prendido. Además, se debe actualizar la temperatura en el LCD.
3. Modificar el sensor de luz. Si el nivel de luz establecido es menor al 90% entonces debe prender la bombilla, y en caso contrario se tiene que apagar. Además, se debe actualizar el porcentaje de luz en el LCD.

Tener en cuenta que cada 250 ms se realiza el senseo, por lo que los actuadores no se actualizarán instantáneamente sino que habrá un retardo intencional.

LINK A PROYECTO: <https://www.tinkercad.com/things/iM3oSQ6kAIO-sancheztomas-granjaautonoma>