

## Sensor DHT para medir temperatura ambiente en Arduino uno

Grado sugerido: Décimo

Deiber Andrés Aldana Pulido

*Publicado en el Banco Virtual de Recursos de Colombia Programa en el año 2025.*



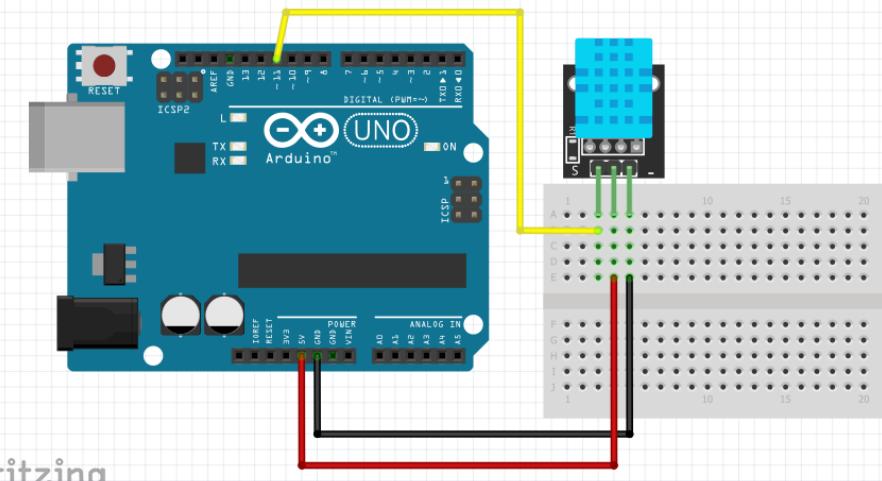
Este material se comparte bajo la licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0). Puede copiar y redistribuir el material en cualquier medio o formato, siempre que dé el crédito adecuado al autor, no lo use con fines comerciales, y no remezcle, transforme o cree a partir del material.

Para más información, consulte la licencia completa en [Deed - Attribution-NonCommercial-NoDerivatives 4.0 International - Creative Commons](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Para contactar al autor/a de este recurso, escriba a: [ingaldana569@gmail.com](mailto:ingaldana569@gmail.com)

## SENSOR DHT11 PARA MEDIR TEMPERATURA AMBIENTE EN ARDUINO UNO

<b>Aprendizaje(s) esperado(s)</b>	<ul style="list-style-type: none"> <li>• Conectar adecuadamente el sensor DHT11 y un LED a la placa Arduino UNO.</li> <li>• Programar la lectura de temperatura desde el sensor y visualizarla en el monitor serial.</li> <li>• Implementar una condición lógica que active un LED cuando la temperatura supere un valor umbral.</li> </ul>
<b>Materiales requeridos</b>	<ul style="list-style-type: none"> <li>• Sensor DHT11</li> <li>• Arduino UNO con cable USB</li> <li>• 1 led</li> <li>• 1 Resistencia de 220Ω</li> <li>• Protoboard</li> <li>• Cables de conexión</li> <li>• Computador con IDE de Arduino</li> </ul>
<b>Conocimientos previos requeridos</b>	<ul style="list-style-type: none"> <li>• Identificar los pines de entrada y salida de la placa Arduino.</li> <li>• Conectar componentes en protoboard.</li> <li>• Cargar y verificar código en el IDE de Arduino.</li> </ul>
<b>Actividad(es) a desarrollar</b>	<b>Tiempo estimado</b>
<i>Indique las acciones que realizarán el/la docente y sus estudiantes y las indicaciones si el trabajo se debe realizar de forma individual, en parejas o grupal.</i>	<i>Minutos o porcentaje</i>
<p><b>1. Inicio (Montaje y configuración)</b></p> <p>✓ El DHT11 es un sensor de humedad relativa y temperatura de bajo costo y de media precisión a un bajo precio. La salida suministrada es de tipo digital utilizando solamente 1 pin de datos.</p> <p>Antes de usarlo, se debe completar el proceso de instalación de la librería que se ilustra en el <b>Anexo D</b>. En caso de que ya esté instalada, este procedimiento no es necesario</p> <p>✓ Construir el siguiente circuito electrónico:</p>	<b>20 minutos</b>



mitzing

**Nota:** tener en cuenta que existen tarjetas de sensor DHT11 que tienen una distribución diferente de los pines, es decir, en otro orden.

- ✓ Configurar el IDE de Arduino para trabajar con la tarjeta Arduino UNO (**Anexo E**)

## 2. Desarrollo

- Escribir el siguiente código en el IDE de Arduino:

```
//Fecha:
//Autores:
//Descripción:

#include "DHT.h"           // Incluye la librería DHT para poder usar sensores de temperatura y humedad como el DHT11

#define DHTPIN 11           // Define el pin digital 11 como la entrada de datos del sensor DHT11
#define DHTTYPE DHT11        // Especifica que el tipo de sensor utilizado es el DHT11

int led1 = 7;               // Declara el pin digital 7 para conectar un LED que se encenderá si la temperatura es alta

DHT dht(DHTPIN, DHTTYPE); // Crea un objeto "dht" para manejar las lecturas del sensor con el pin y tipo definidos

float temperatura;         // Variable tipo float para almacenar el valor leído de la temperatura

void setup() {
  Serial.begin(9600);      // Inicializa la comunicación serial a 9600 baudios (velocidad estándar)
  Serial.println("Iniciando..."); // Muestra un mensaje inicial en el monitor serial
  dht.begin();              // Inicializa el sensor DHT11
  pinMode(led1, OUTPUT);    // Configura el pin del LED como salida
}

void loop() {
  temperatura = dht.readTemperature(); // Lee la temperatura desde el sensor y la guarda en la variable

  Serial.print("Temperatura: ");      // Imprime el texto en el monitor serial
  Serial.print(temperatura);          // Imprime el valor de temperatura
  Serial.println(" °C");              // Imprime la unidad de medida

  if (temperatura > 32) {           // Condición: si la temperatura es mayor a 32 grados
    Serial.println("Alerta!! temperatura alta"); // Mensaje de advertencia por calor
  }

  delay(2000); // Espera 2 segundos antes de la siguiente lectura para evitar lecturas constantes
}
```

- Cargar el código y abrir el monitor serial como se indica en el **Anexo F**.
- ✓ Analizar la estructura del código y describir su funcionamiento. Responde las siguientes preguntas:

80 minutos

- ¿Cada cuánto tiempo se realiza la lectura de la temperatura?
  - ¿Qué sucede si la temperatura es superior a 32 grados?
  - ¿se activa alguna salida digital en caso de que la temperatura sea superior a 32 grados?
- ✓ Acerque hielo al sensor (sin que se moje) y verifique la lectura en el monitor serial. ¿Cuál es el valor mínimo leído?
- ✓ Acerque una vela encendida al sensor (sin que lo toque) y verifique la lectura en el monitor serial. ¿Cuál es el valor máximo leído?

**✓ Ejercicio 1:**

Modificar el circuito y el algoritmo de tal forma que cumpla con los siguientes requisitos:

- Se debe agregar un led conectado al pin 7 del arduino
- El led se debe encender cuando la temperatura supere el valor máximo permitido de 32 grados
- El led se debe apagar cuando la temperatura es inferior a 32 grados.

**3. Cierre**

Para finalizar la sesión, invite a los estudiantes a reflexionar sobre la utilidad del sensor DHT11 en contextos fuera del aula. Por ejemplo:

- ✓ ¿Cómo se podría aplicar este sensor en un sistema de control de temperatura para invernaderos, refrigeradores o espacios domésticos?
- ✓ ¿Qué otras variables ambientales podrían medirse con sensores similares?
- ✓ ¿Qué impacto tiene automatizar estos procesos en la eficiencia energética y el cuidado de recursos como el agua?

Proponga una breve lluvia de ideas en grupo para imaginar un proyecto en el que se integre este sensor con otros componentes (como ventiladores, pantallas o módulos Wi-Fi), orientado a resolver un problema local (ej. calor extremo, protección de cultivos, monitoreo ambiental).

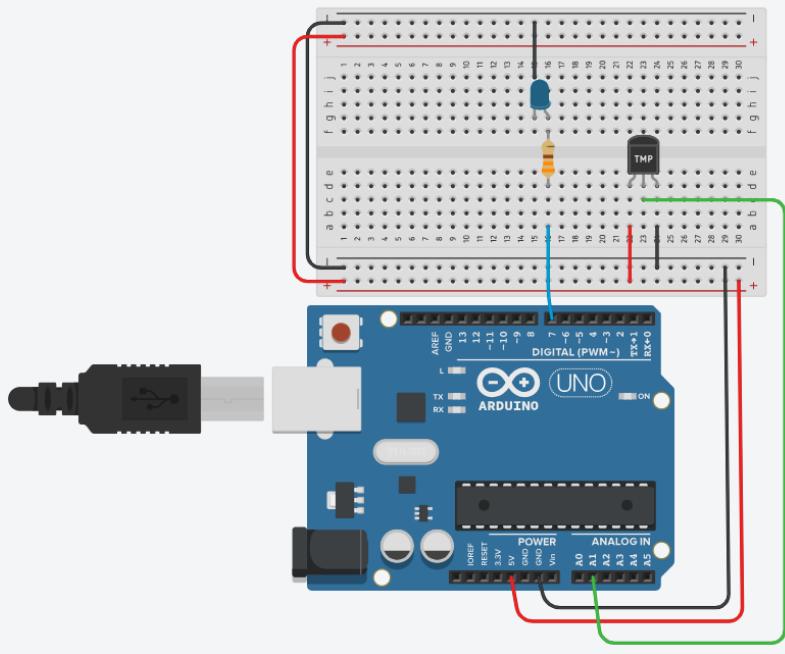
**20 minutos**

**Adaptaciones**

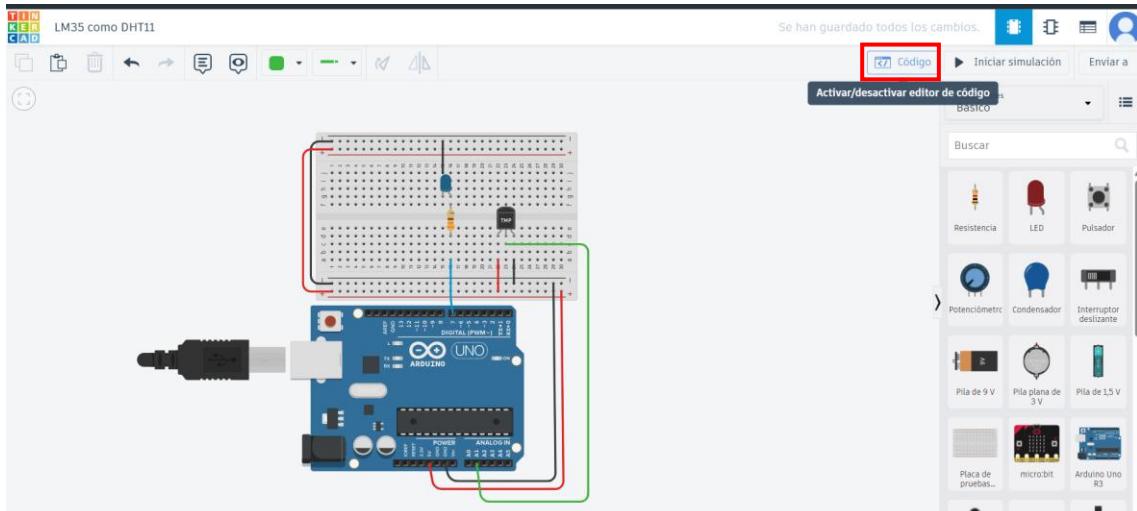
**• Para estudiantes sin materiales físicos:**

Los estudiantes que no dispongan del hardware pueden utilizar el simulador de circuitos electrónicos en la plataforma Tinkercad, siguiendo las instrucciones del **Anexo G**.

1. Tinkercad no cuenta como el sensor DHT11 para realizar simulaciones, por lo que usaremos el sensor LM35 que es un sensor análogo para la medición de la temperatura ambiente. Realizar el circuito electrónico

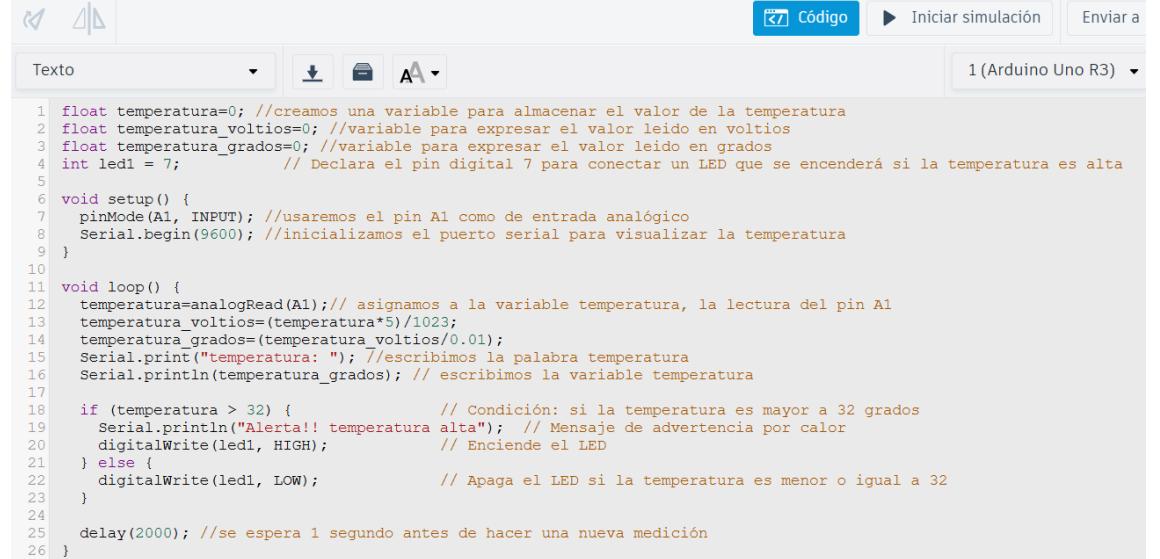


## 2. Seleccionar la opción “Código”



## 3. Ingresar el programa correspondiente.

Nota: el código anterior ya contiene la solución al ejercicio 1 propuesto en el desarrollo de la práctica.

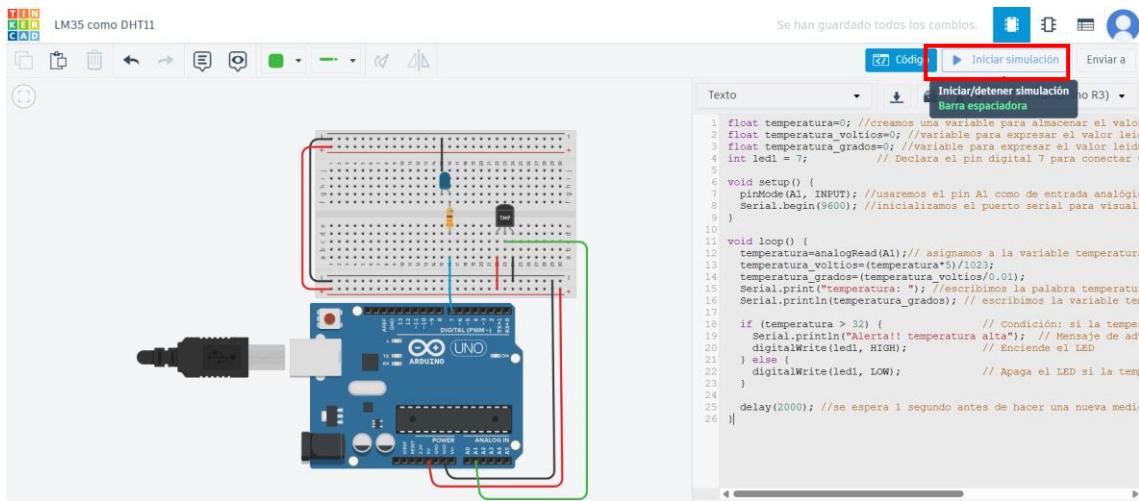


```

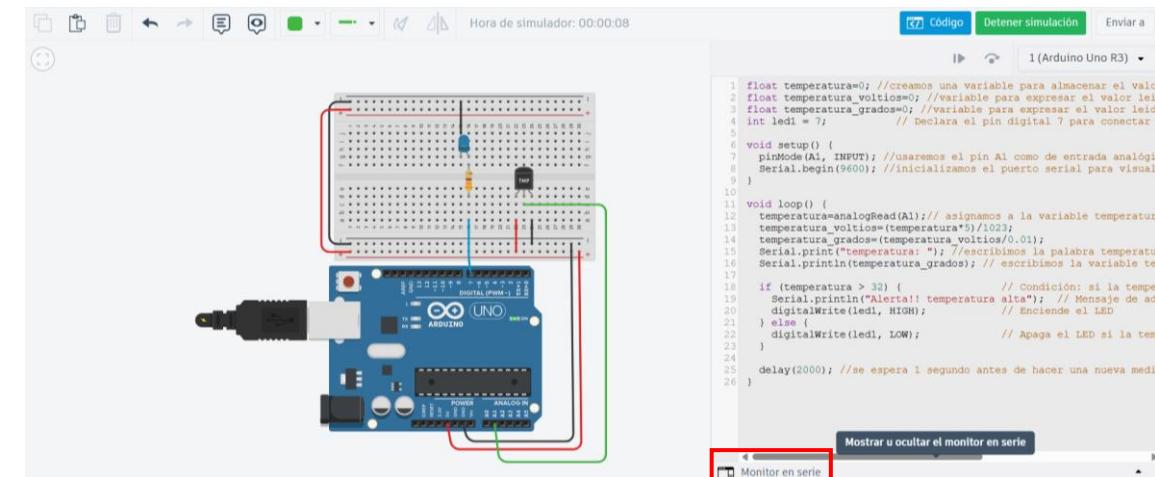
1 float temperatura=0; //Creamos una variable para almacenar el valor de la temperatura
2 float temperatura_voltios=0; //variable para expresar el valor leido en voltios
3 float temperatura_grados=0; //variable para expresar el valor leido en grados
4 int led1 = 7; // Declara el pin digital 7 para conectar un LED que se encenderá si la temperatura es alta
5
6 void setup() {
7   pinMode(A1, INPUT); //usaremos el pin A1 como de entrada analógico
8   Serial.begin(9600); //inicializamos el puerto serial para visualizar la temperatura
9 }
10
11 void loop() {
12   temperatura=analogRead(A1); // asignamos a la variable temperatura, la lectura del pin A1
13   temperatura_voltios=(temperatura*5)/1023;
14   temperatura_grados=(temperatura_voltios/0.01);
15   Serial.print("temperatura: "); //escribimos la palabra temperatura
16   Serial.println(temperatura_grados); // escribimos la variable temperatura
17
18   if (temperatura > 32) { // Condición: si la temperatura es mayor a 32 grados
19     Serial.println("Alerta!! temperatura alta"); // Mensaje de advertencia por calor
20     digitalWrite(led1, HIGH); // Enciende el LED
21   } else {
22     digitalWrite(led1, LOW); // Apaga el LED si la temperatura es menor o igual a 32
23   }
24
25   delay(2000); //se espera 1 segundo antes de hacer una nueva medición
26 }

```

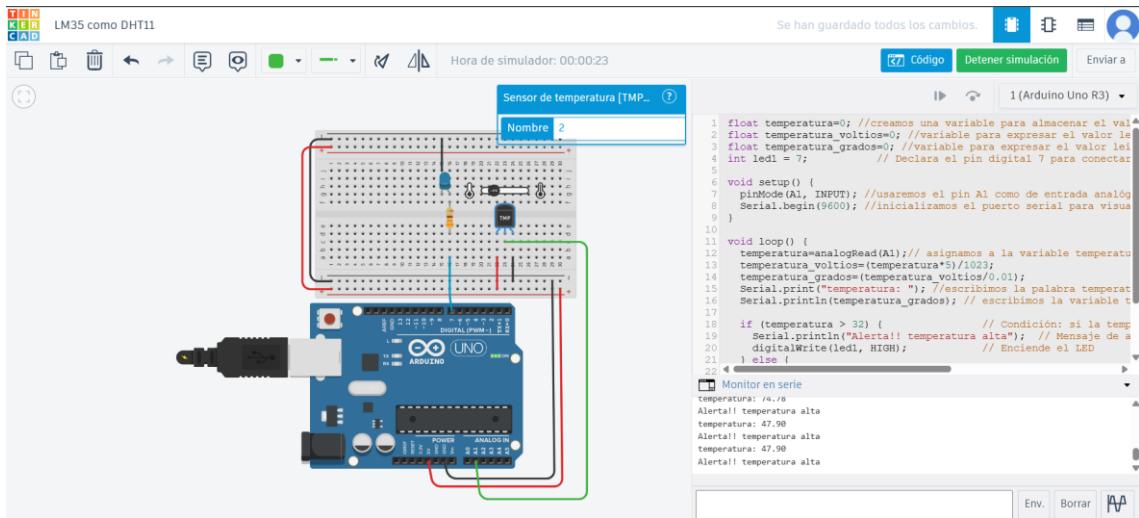
#### 4. Presionar el botón para comenzar la simulación.



#### 5. Acceder al Monitor Serial para observar la transmisión de datos.



## 6. Probar distintos valores de temperatura ambiente y evaluar cómo responde el circuito.



- Baja conectividad:** En lugares con acceso limitado a internet, se puede trabajar con esquemas impresos del circuito y pseudocódigo. Los estudiantes pueden simular el funcionamiento en papel utilizando tarjetas que representen los componentes y describiendo con flechas y anotaciones cómo se comportaría el sistema ante diferentes temperaturas.
- Grupos numerosos:** Para optimizar el tiempo y los recursos, se recomienda organizar estaciones de trabajo por tareas: una para el diseño del circuito, otra para la escritura del código, una para la simulación y otra para registrar los resultados. También se pueden asignar roles rotativos dentro de cada equipo, como programador, técnico de montaje, observador y analista.
- Enfoque intercultural:** Se pueden incluir ejemplos de aplicación del sensor en contextos locales, como el monitoreo de temperatura en cultivos o en viviendas sin ventilación adecuada. También se puede invitar a los estudiantes a diseñar una propuesta de solución a un problema ambiental o social de su comunidad utilizando este tipo de sensores.

### Actividades evaluativas

Para evaluar los aprendizajes esperados, el docente puede apoyarse en tres evidencias clave:

- Observación del montaje físico del sensor y LED.
- Revisión del código funcional y su lógica.
- Explicación oral del funcionamiento del sistema de alerta.

### Referencias

Arduino.cl. (2025). *Principales partes de un Arduino*.  
[https://arduino.cl/principales-partes-de-un-arduino/?srltid=AfmBOoqUF-vw7m4c--W6rRyaCGpqT4ZEsQy6AFxOPM\\_BGsm-HMwYKiI0](https://arduino.cl/principales-partes-de-un-arduino/?srltid=AfmBOoqUF-vw7m4c--W6rRyaCGpqT4ZEsQy6AFxOPM_BGsm-HMwYKiI0)

## ANEXOS

### ANEXO A. tarjeta Arduino UNO – Distribución de partes

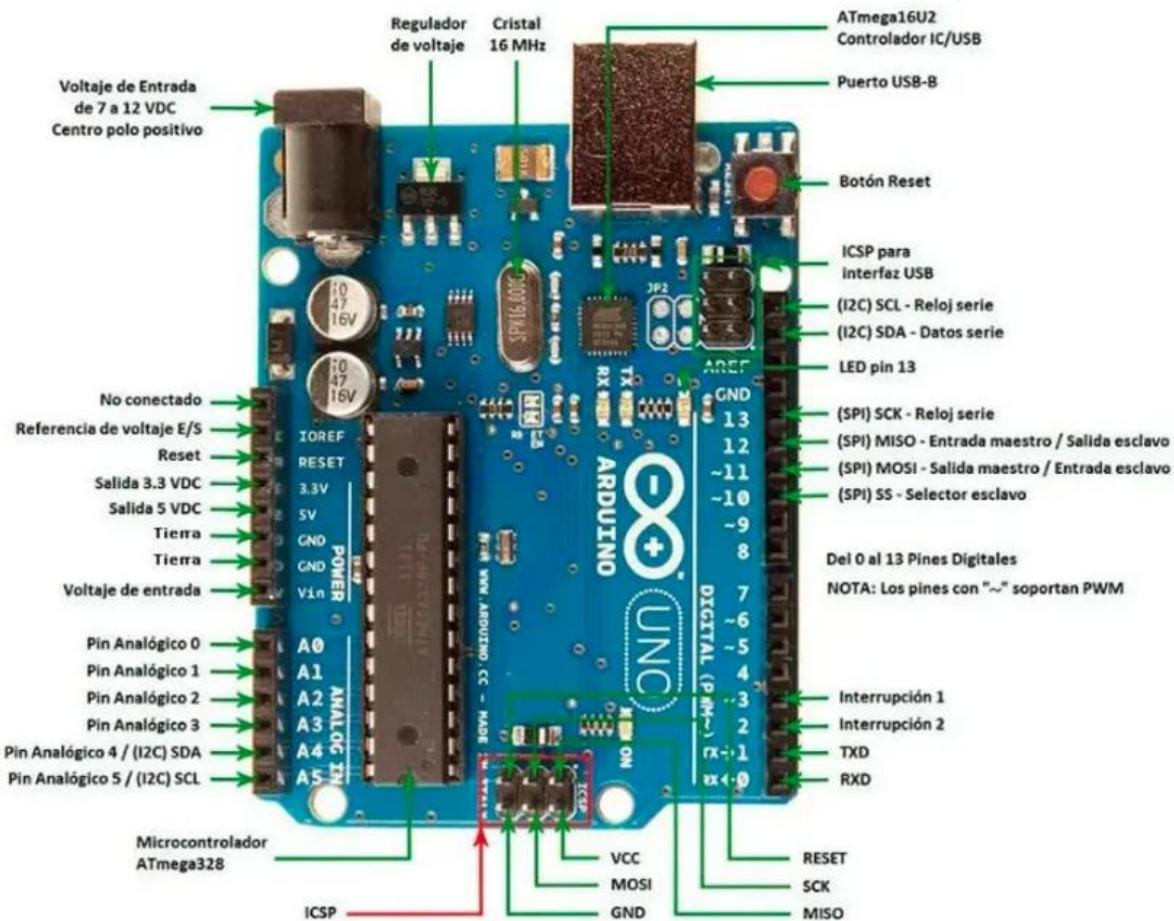
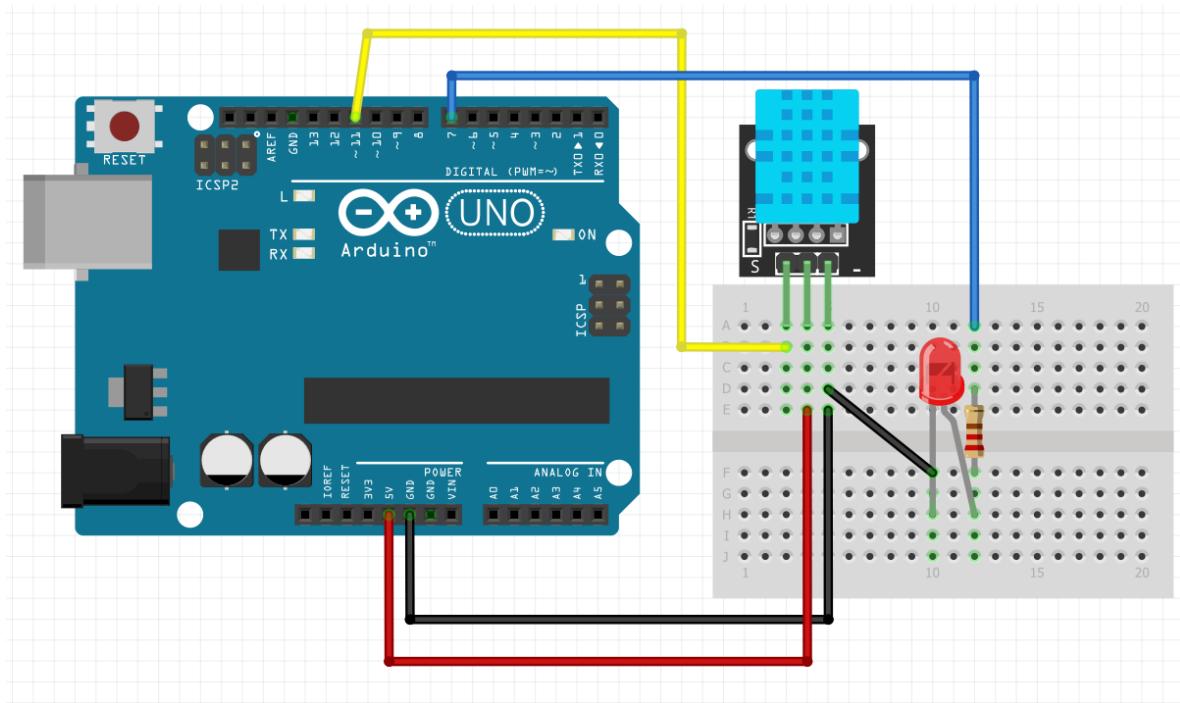


Imagen tomada de (Arduino.cl, 2025)

## ANEXO B. Circuito solución al Ejercicio 1



## ANEXO C. Algoritmo solución al Ejercicio 1

```
#include "DHT.h"          // Incluye la librería DHT para poder usar sensores de temperatura y humedad como el DHT11

#define DHTPIN 11          // Define el pin digital 11 como la entrada de datos del sensor DHT11
#define DHTTYPE DHT11        // Especifica que el tipo de sensor utilizado es el DHT11

int led1 = 7;              // Declara el pin digital 7 para conectar un LED que se encenderá si la temperatura es alta

DHT dht(DHTPIN, DHTTYPE); // Crea un objeto "dht" para manejar las lecturas del sensor con el pin y tipo definidos

float temperatura;        // Variable tipo float para almacenar el valor leído de la temperatura

void setup() {
  Serial.begin(9600);      // Inicializa la comunicación serial a 9600 baudios (velocidad estándar)
  Serial.println("Iniciando..."); // Muestra un mensaje inicial en el monitor serial
  dht.begin();              // Inicializa el sensor DHT11
  pinMode(led1, OUTPUT);    // Configura el pin del LED como salida
}

void loop() {
  temperatura = dht.readTemperature(); // Lee la temperatura desde el sensor y la guarda en la variable

  Serial.print("Temperatura: ");      // Imprime el texto en el monitor serial
  Serial.print(temperatura);          // Imprime el valor de temperatura
  Serial.println(" °C");              // Imprime la unidad de medida

  if (temperatura > 32) {           // Condición: si la temperatura es mayor a 32 grados
    Serial.println("Alerta!! temperatura alta"); // Mensaje de advertencia por calor
    digitalWrite(led1, HIGH);         // Enciende el LED
  } else {
    digitalWrite(led1, LOW);         // Apaga el LED si la temperatura es menor o igual a 32
  }

  delay(2000); // Espera 2 segundos antes de la siguiente lectura para evitar lecturas constantes
}
```

## ANEXO D. CONFIGURACIÓN DE LIBRERÍA DHT11



## ANEXO E. CONFIGURACIÓN IDE DE ARDUINO



**ANEXO F. PROCESO DE CARGA DEL CÓDIGO EN LA TARJETA ARDUINO UNO**



**ANEXO G. CREACIÓN DE CUENTA TINKERCAD PARA CIRCUITOS ELECTRÓNICOS**

