



## **Actividad |3| Pantalla de Temperatura.**

**Internet de las Cosas.**

---

**Ingeniería en Desarrollo de Software.**



TUTOR: Marco Alonso Rodríguez Tapia.

ALUMNO: Ramón Ernesto Valdez Felix.

FECHA: 19/02/2026.

<b>Introducción.....</b>	<b>3</b>
<b>Descripción.....</b>	<b>3</b>
<b>Justificación.....</b>	<b>4</b>
<b>Desarrollo.....</b>	<b>4</b>
<b>Armado del Circuito.....</b>	<b>5</b>
<b>Codificación.....</b>	<b>7</b>
<b>Emulación del Circuito.....</b>	<b>9</b>
<b>Conclusion.....</b>	<b>11</b>
<b>Referencias.....</b>	<b>11</b>

## Introducción.

En esta actividad final de la materia de Internet de las Cosas (IoT), consiste en el establecer el marco teórico y práctico para el desarrollo de un sistema de monitoreo térmico dentro del ecosistema del Internet de las Cosas. Basándose en los conceptos analizados en los videos 3 y 4 de la materia, la actividad se centra en la integración de hardware y software mediante la plataforma Tinkercad. El objetivo primordial es diseñar un circuito capaz de capturar datos analógicos del entorno a través de un sensor de temperatura y transformarlos en información visual legible en una pantalla LCD de 16x2.

Para lograrlo, es fundamental el dominio de la librería LiquidCrystal, la cual permite la comunicación entre la placa Arduino y el display. El proceso técnico involucra el uso de variables de tipo float para garantizar la precisión de los datos, así como la implementación de funciones críticas como analogRead y map para la conversión de señales. Esta práctica no solo refuerza el conocimiento sobre el cableado de componentes como el potenciómetro y resistencias, sino que también sienta las bases para crear sistemas de alerta automatizados y dispositivos inteligentes interconectados.

## Descripción.

En esta actividad final de Internet de las Cosas (IoT), se centra en la convergencia entre el hardware analógico y la programación digital dentro del marco del Internet de las Cosas (IoT). Utilizando la plataforma Tinkercad, se desarrolla un sistema de monitoreo que integra componentes esenciales como una placa Arduino, un sensor de temperatura y una pantalla LCD de 16x2 para la visualización de datos. El núcleo técnico del proyecto reside en la correcta importación de la librería LiquidCrystal y la configuración de pines específicos para permitir la comunicación entre el microcontrolador y la interfaz visual.

El proceso implica la creación de variables de tipo float para manejar la precisión térmica, empleando la función analogRead combinada con map para traducir las señales eléctricas en valores de temperatura legibles. Al inicializar el mensaje "Temperatura" en el void setup y procesar los cálculos en el void loop,

el estudiante aprende a transformar variables físicas en información digital. Esta práctica es fundamental para comprender cómo los dispositivos inteligentes interactúan con su entorno para la toma de decisiones automatizadas.

## Justificación.

La realización de esta actividad se justifica por la necesidad de comprender la interacción entre el mundo físico y digital, un concepto central en el Internet de las Cosas (IoT). Al integrar un sensor de temperatura con una pantalla LCD, el estudiante desarrolla habilidades críticas para la digitalización de variables analógicas. El uso de herramientas virtuales como Tinkercad facilita la experimentación segura con componentes como la placa Arduino, el potenciómetro y resistencias, permitiendo una comprensión profunda del flujo eléctrico y la lógica de control.

Desde el punto de vista técnico, la implementación de la librería LiquidCrystal y el uso de funciones como MAP y analogRead son fundamentales para el procesamiento de datos en tiempo real. Esta práctica no solo enseña a programar variables tipo float para obtener precisión en grados Celsius, sino que también prepara al alumno para diseñar sistemas de monitoreo autónomos. En última instancia, esta actividad es el primer paso para crear infraestructuras inteligentes capaces de visualizar información crítica para la toma de decisiones en entornos industriales y domésticos.

## Desarrollo.

El desarrollo de esta actividad se centra en la transición del aprendizaje teórico a la implementación práctica de un sistema de control básico dentro del ecosistema del Internet de las Cosas. Al simular el circuito, se obtiene un sistema funcional que replica el comportamiento de un detector de temperatura real y los muestra en su pantalla, demostrando la capacidad de automatizar procesos cotidianos mediante la programación de microcontroladores.

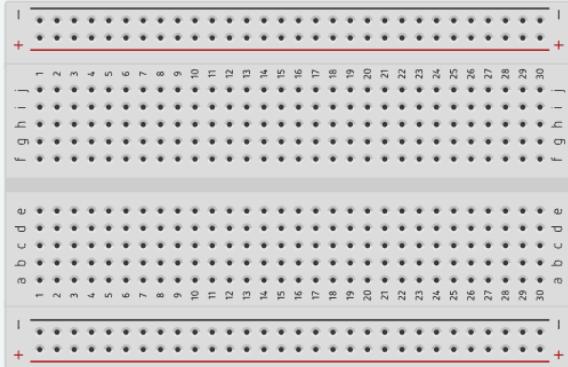
[Link: GitHub.](#)

[Link: Actividad.](#)

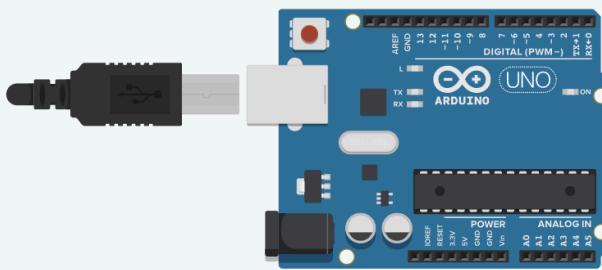
# Armado del Circuito.

Utilizamos los componentes básicos de la herramienta de trabajo tinkerCAD para realizar la actividad de la creación de una pantalla de temperatura.

## Circuitos a utilizar: Pantalla de temperatura.



**Placa de pruebas (Breadboard):** Es la base blanca con agujeros que permite conectar los componentes entre sí sin necesidad de soldar. Las líneas horizontales y verticales están conectadas internamente para compartir electricidad.



**Placa Arduino Uno:** Es el "cerebro" programable. Recibe el código que escribes y envía señales eléctricas a través de sus pines para controlar los LEDs.



**TMP36:** Mide la temperatura ambiente. Envía una señal de voltaje variable al pin A0 del Arduino proporcional al calor detectado.



**Resistencias ( 220 ohms):** Son esos pequeños cilindros con bandas de colores. Su función es limitar la cantidad de corriente que llega a los LEDs para evitar que se quemen.

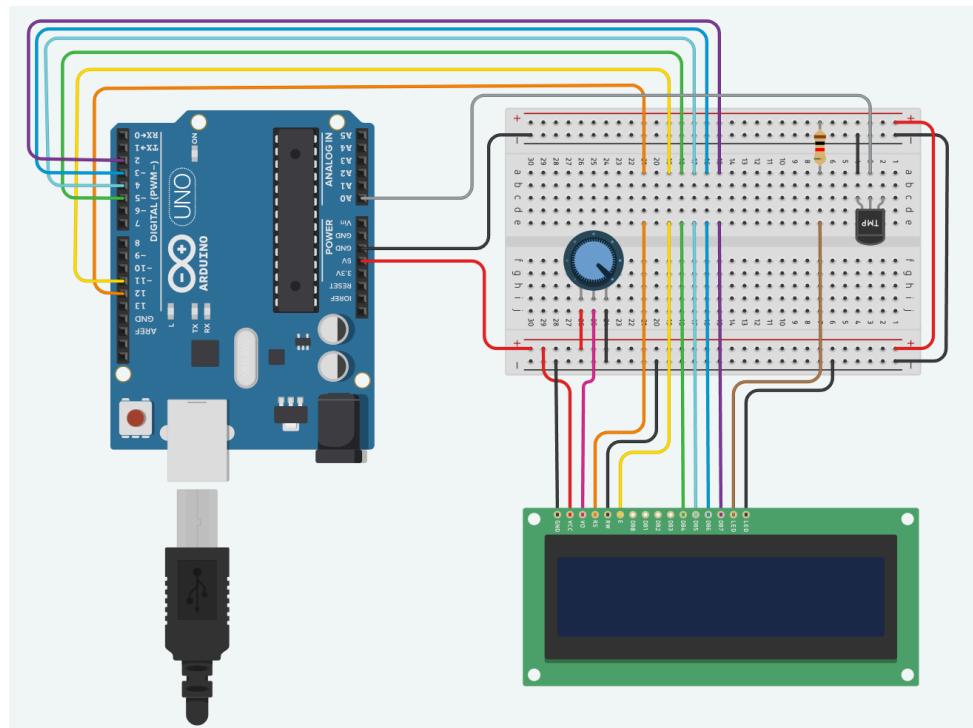


**Pantalla LCD 16x2:** Interfaz visual que muestra los datos procesados. Tiene capacidad para mostrar 16 caracteres en 2 líneas.



**Potenciómetro:** Actúa como una resistencia variable. En este circuito, se usa para ajustar el contraste de las letras en el LCD.

**Cables de conexión (Jumpers):** Los hilos de colores que transportan la energía y los datos entre la placa Arduino y la placa de pruebas. Con todos estos circuitos básicos realizamos el armado de actividad (pantalla de temperatura).



# Codificación.

En este punto de la actividad realizaremos la documentación de la codificación de la pantalla de temperatura que nos pide realizar la materia de internet de las cosas como primer actividad y daremos un explicación breve de lo que realiza el código.

## Codificación de la pantalla de temperatura.

```
#include <LiquidCrystal.h>

int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

const int sensorPin = A0;

float temperaturaC;

float temperaturaF;

void setup() {

    lcd.begin(16, 2);

    lcd.print("Temperatura");

    delay(1500);
```

### 1. Preparación de la Pantalla (Librería y Pines)

- **#include <LiquidCrystal.h>**: Importa la librería necesaria para controlar pantallas LCD.
- **Variables int**: Se definen los pines digitales del Arduino (12, 11, 5, 4, 3, 2) que se conectan físicamente a la pantalla.
- **Objeto lcd**: Configura la pantalla con esos pines específicos para que el código sepa a dónde enviar la información.

### 2. Configuración Inicial (void setup)

- **lcd.begin(16, 2)**: Indica que estamos usando una pantalla de 16 columnas y 2 filas.
- **lcd.print("Temperatura")**: Al

<pre> lcd.clear(); }  void loop() {      long     valorMapeado      = map(analogRead(sensorPin), 0, 1023, -5000, 45000);      temperaturaC = valorMapeado / 100.0;      temperaturaF = (temperaturaC * 9.0 / 5.0) + 32.0;      lcd.setCursor(0, 0);      lcd.print("C: ");      lcd.print(temperaturaC, 1);      lcd.print(" Grad");      lcd.setCursor(0, 1); </pre>	<p>encender el circuito, aparece este mensaje de bienvenida por 1.5 segundos antes de empezar a medir.</p> <p><b>3. Procesamiento de Datos (void oop)</b></p> <p>El código repite los siguientes pasos cada segundo:</p> <ol style="list-style-type: none"> <li><b>Lectura y Mapeo:</b> Usa <code>analogRead</code> para obtener un valor del sensor (entre 0 y 1023) y la función <code>map</code> lo transforma a un rango numérico que representa la temperatura multiplicada por 100 para no perder precisión.</li> <li><b>Cálculo de Celsius:</b> Divide el valor mapeado entre 100.0 para obtener la temperatura real en una variable <code>float</code> (con decimales).</li> <li><b>Conversión a Fahrenheit:</b> Aplica la fórmula matemática <math>\text{F} = (\text{C} \times 1.8) + 32</math> para obtener la temperatura en la escala estadounidense.</li> <li><b>Visualización en la Pantalla</b></li> </ol>
---	---

```

lcd.print("F: ");

lcd.print(temperaturaF, 1);

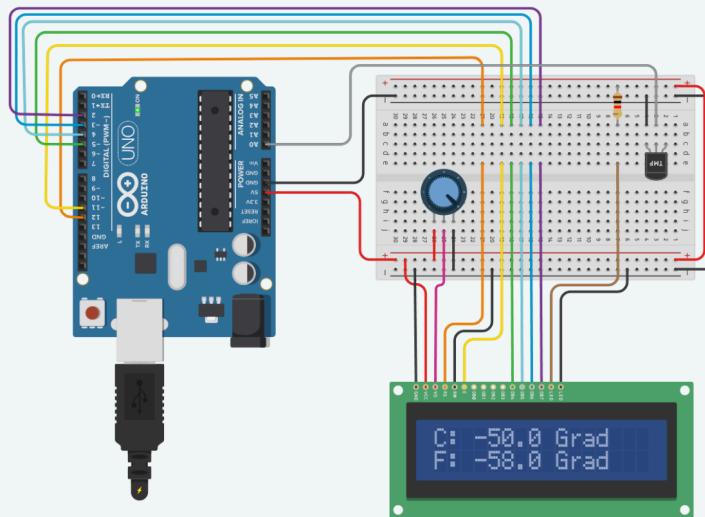
lcd.print(" Grad");

delay(1000);

}

```

- **lcd.setCursor:** Posiciona el "cursor" de escritura. La fila 0 (arriba) muestra los grados Celsius y la fila 1 (abajo) los grados Fahrenheit.
- **lcd.print:** Envía el texto y los valores calculados a la pantalla para que el usuario pueda leerlos.



```

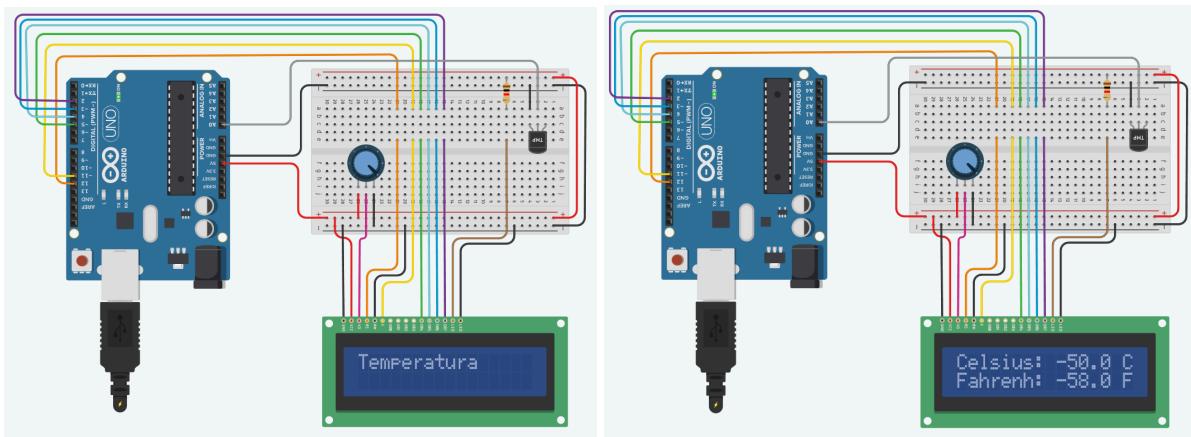
1 //include <LiquidCrystal.h> // Exportar la función LiquidCrystal
2 // Variables de tipo entero para los pines del LCD
3 // d4 = 12, d5 = 11, d6 = 5, d7 = 3, d7 = 2;
4 //lcd.begin(d4, d5, d6, d7, d8, d9);
5 LiquidCrystal lcd(r3, d4, d5, d6, d7, d8);
6
7 const int sensorPin = A0;
8 float temperaturaC; // Variable tipo float para el sensor
9 float temperaturaF;
10
11 void setup() {
12   // Configurar pixeles 16x2
13   lcd.begin(16, 2);
14   // Muestra el mensaje inicial
15   lcd.print("Temperatura");
16   delay(1500);
17   lcd.clear();
18 }
19
20 void loop() {
21   // Uso de la función MAP con analogRead
22   // Mapamos el rango de lectura (0-1023) a los valores crudos del TMP36 (-50 a 125
23   float valorMapeado = map(analogRead(sensorPin), 0, 1023, -5000, 45000);
24
25   // Dividir el valor entre 100 para obtener Celsius
26   temperaturaC = valorMapeado / 100.0;
27
28   // Calcular Fahrenheit
29   temperaturaF = (temperaturaC * 9.0 / 5.0) + 32.0;
30
31   // Manda a Imprimir en la pantalla LCD
32   lcd.setCursor(0, 0);
33   lcd.print("C: ");
34   lcd.print(temperaturaC, 1);
35   lcd.print(" F:");
36
37   lcd.setCursor(0, 1);
38   lcd.print("F: ");
39   lcd.print(temperaturaF, 1);
40   lcd.print(" Grad");
41
42   delay(1000);
43 }

```

## Emulación del Circuito.

Al emular este código en una plataforma como Tinkercad, estás ejecutando un programa que convierte instrucciones lógicas en señales eléctricas reales. Aquí se tiene la emulación de los circuitos en su ejecución donde al girar el potenciómetro ilumina la pantalla donde muestra los grados Celsius y Fahrenheit en la Pantalla LCD 16x2, aplicado en la lógica de la codificación la pantalla de temperatura.

## Prueba de emulación de actividad de pantalla de temperatura.



< Gallery    Private



### Sizzling Wluff

React 0

Circuit by  
Ramón Valdez

Tinker this

Share to Classroom

Copy link

Design is visible only to you.

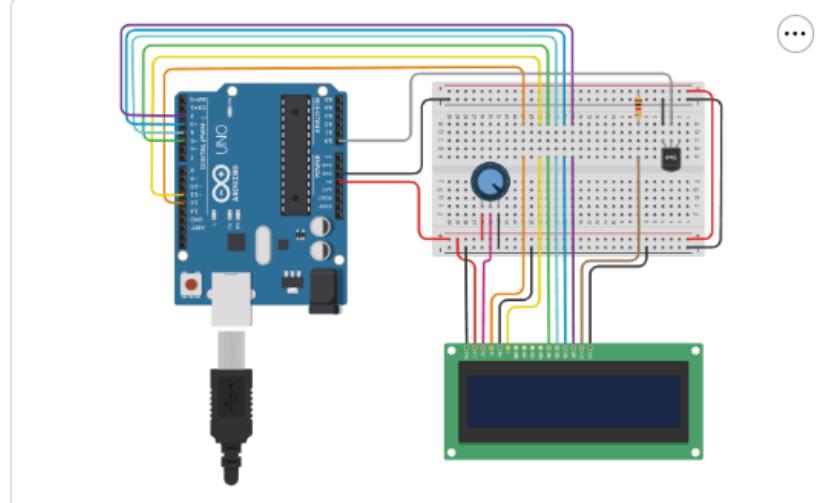
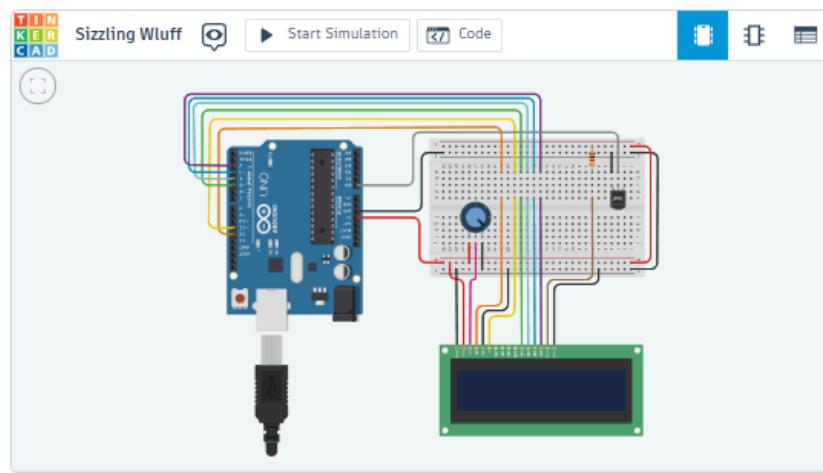
[Change visibility to share](#)

Edited February 19, 2026

Created February 19, 2026



Report content



<https://www.tinkercad.com/things/0DNpFMY0zda-sizzling-wluff>

## Conclusion.

En conclusión: La creación de un monitor de temperatura con pantalla LCD en el ámbito del Internet de las Cosas (IoT) es una actividad de vital importancia, ya que representa la base de los sistemas de control ambiental inteligentes. En el campo laboral, dominar la integración de sensores analógicos y visualizadores digitales permite diseñar soluciones de monitoreo para centros de datos, almacenes de alimentos o procesos industriales donde el control térmico es crítico para evitar pérdidas materiales.

En la vida cotidiana, esta tecnología es el núcleo de la domótica, permitiendo que los hogares se vuelvan más eficientes al gestionar sistemas de calefacción o aire acondicionado de forma automática. La relevancia de esta práctica radica en aprender a transformar variables físicas del entorno en información digital legible y accionable. Al programar conversiones entre Celsius y Fahrenheit, el profesional desarrolla competencias para crear dispositivos globales y versátiles, esenciales en una infraestructura tecnológica cada vez más conectada y dependiente de la toma de decisiones basada en datos precisos.

## Referencias.

*Google. (n.d.). Gemini. Retrieved February 20, 2026, from*

*<https://gemini.google.com/>*

*Tinkercad. (n.d.). Tinkercad. Retrieved February 20, 2026, from*

*<https://www.tinkercad.com/dashboard/designs/all>*