

## Actividad | 3 | Pantalla de

## Temperatura

## Internet de las Cosas

---

## Ingeniería en Desarrollo de

## Software



TUTOR: Marco Alonso Rodríguez Tapia

ALUMNO: Eduardo Monroy Hernández

FECHA: 25-10-2025

# Introducción

En la actualidad, la tecnología forma parte de nuestra vida diaria y cada vez más objetos pueden conectarse entre sí para hacernos las tareas más fáciles. A esto se le conoce como Internet de las Cosas. Gracias a esta tecnología, es posible crear proyectos donde los dispositivos recolectan información del entorno, la procesan y la muestran de una manera sencilla. En esta actividad aprenderás a realizar un circuito con Arduino, un sensor de temperatura y una pantalla LCD, con el propósito de que el sensor mida la temperatura y esta se muestre en grados Celsius en la pantalla.

Antes de comenzar, se nos recomienda que es importante ver los videos 3 y 4 de la materia Internet de las Cosas, ya que ahí se explica paso a paso cómo conectar y programar los componentes que se usarán. El objetivo de este ejercicio es que comprendemos cómo funcionan los sensores y cómo se puede mostrar la información que captan a través de una pantalla.

Para armar el circuito en Tinkercad, necesitaremos algunos elementos como una placa Arduino, una pantalla LCD, un sensor de temperatura, una resistencia, un potenciómetro y una placa de pruebas. En el programa, se utilizará la función LiquidCrystal, que permite controlar la pantalla LCD. También se crearán variables que guardarán los valores de temperatura, y se usarán funciones como analogRead y MAP para transformar los datos que envía el sensor en valores reales.

Finalmente, el sistema mostrará en pantalla el mensaje “Temperatura” junto con el valor detectado, permitiendo ver los cambios en tiempo real. Esta actividad te ayudará a entender cómo la programación y la electrónica pueden trabajar juntas para mostrar información útil del entorno.

## Descripción

En esta actividad aprenderemos a crear un proyecto usando Tinkercad el objetivo es construir un sistema que mida la temperatura del ambiente y la muestre en una pantalla LCD, usando una placa Arduino. De esta forma veremos en tiempo real cuántos grados Celsius hay, tal como lo hacen los termómetros digitales.

Antes de empezar, es importante que miremos los videos 3 y 4 de la materia Internet de las Cosas, ya que ahí se explica cómo conectar los componentes y cómo programarlos para que funcionen correctamente. Una vez revisado los videos, deberemos reunir los siguientes elementos en el simulador de Tinkercad: una placa Arduino, una pantalla LCD, un sensor de temperatura, una placa de pruebas pequeña, un potenciómetro y una resistencia.

El primer paso será conectar todos los componentes entre sí. La pantalla LCD irá conectada a la placa Arduino, la cual se encargará de controlar lo que se muestra. Para que la pantalla funcione, en el programa se debe usar una función llamada LiquidCrystal, que sirve para indicarle al Arduino cómo comunicarse con la pantalla.

Después, en el código se crea una variable que guardará el valor de la temperatura que detecte el sensor. En una parte del programa llamada void setup, se configura la pantalla para que pueda mostrar texto y se escribe el mensaje "Temperatura".

Por último, en la parte llamada void loop, el Arduino leerá constantemente los datos del sensor y los transformará en grados Celsius. Luego, mostrará ese valor en la pantalla LCD, actualizándose cada poco segundo. Al finalizar, tendrás un circuito capaz de medir y mostrar la temperatura de manera automática, lo que te ayudará a comprender cómo la programación y la electrónica pueden trabajar juntas en proyectos prácticos y útiles.

## Justificación

El poder realizar esta actividad ha parecido muy interesante porque se permitió entender de una manera práctica cómo funciona la tecnología que usamos todos los días. A través del uso de Arduino, un sensor de temperatura y una pantalla LCD, veremos cómo un circuito puede captar información del entorno en este caso, la temperatura y mostrarla de forma clara en una pantalla.

Antes de hacer este proyecto, no se tenía mucha idea de cómo los dispositivos “inteligentes” logran medir cosas como la temperatura, pero con esta práctica pude verlo paso a paso y comprenderlo mejor.

Consideramos que esta actividad es importante en donde podremos relacionar la programación con la electrónica de una forma sencilla y visual. Al usar Tinkercad, se puede simular el circuito sin necesidad de materiales físicos, lo cual facilita el aprendizaje, especialmente para quienes apenas estamos comenzando en este tema.

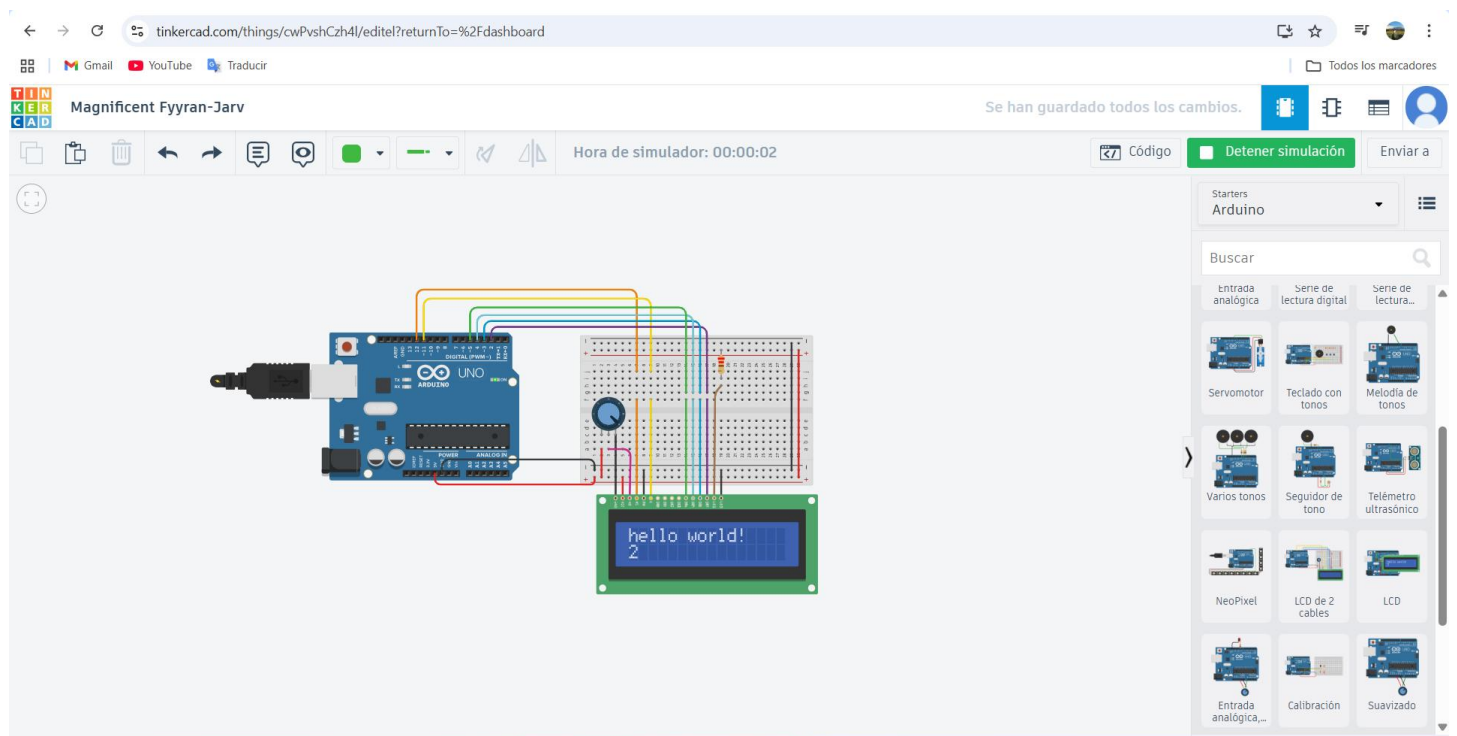
También comprendimos la función de cada componente: el sensor es el que detecta la temperatura, la pantalla LCD es la que muestra el resultado, y el Arduino es como el “cerebro” que procesa la información y hace que todo funcione correctamente.

Otro aspecto que me pareció muy útil fue aprender cómo se escribe el código para que el sistema realice la tarea que queremos. Por ejemplo, usar funciones como `LiquidCrystal` para controlar la pantalla y `analogRead` para leer los datos del sensor me hizo ver cómo el lenguaje de programación permite que los componentes se comuniquen entre sí.

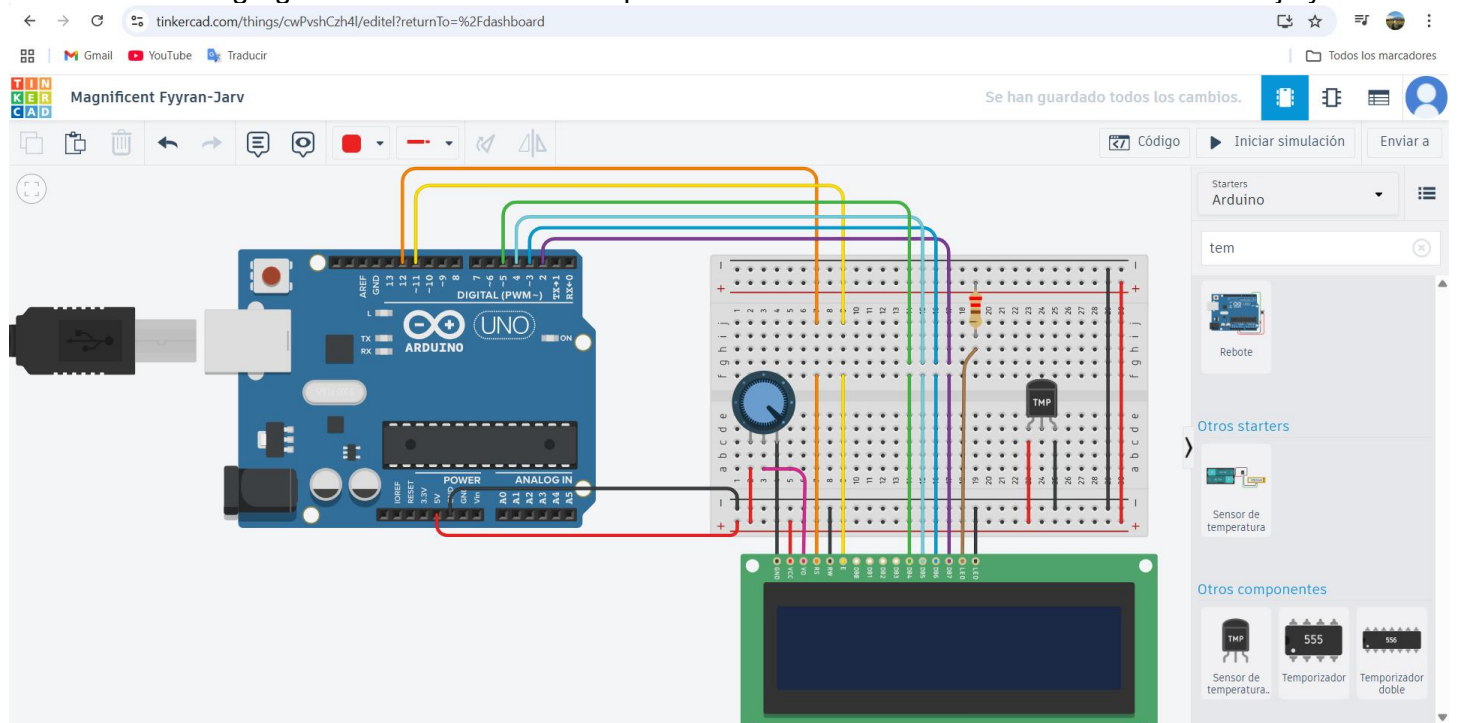
En conclusión, esta actividad no solo me enseñó a medir la temperatura con un sensor, sino también a entender cómo los dispositivos electrónicos pueden automatizar tareas simples. Ahora puedo decir que tengo una idea más clara de cómo funcionan los sistemas del Internet de las Cosas y cómo se aplican en la vida real, desde un termómetro digital hasta otros aparatos más avanzados que usamos a diario.

# Armado del circuito

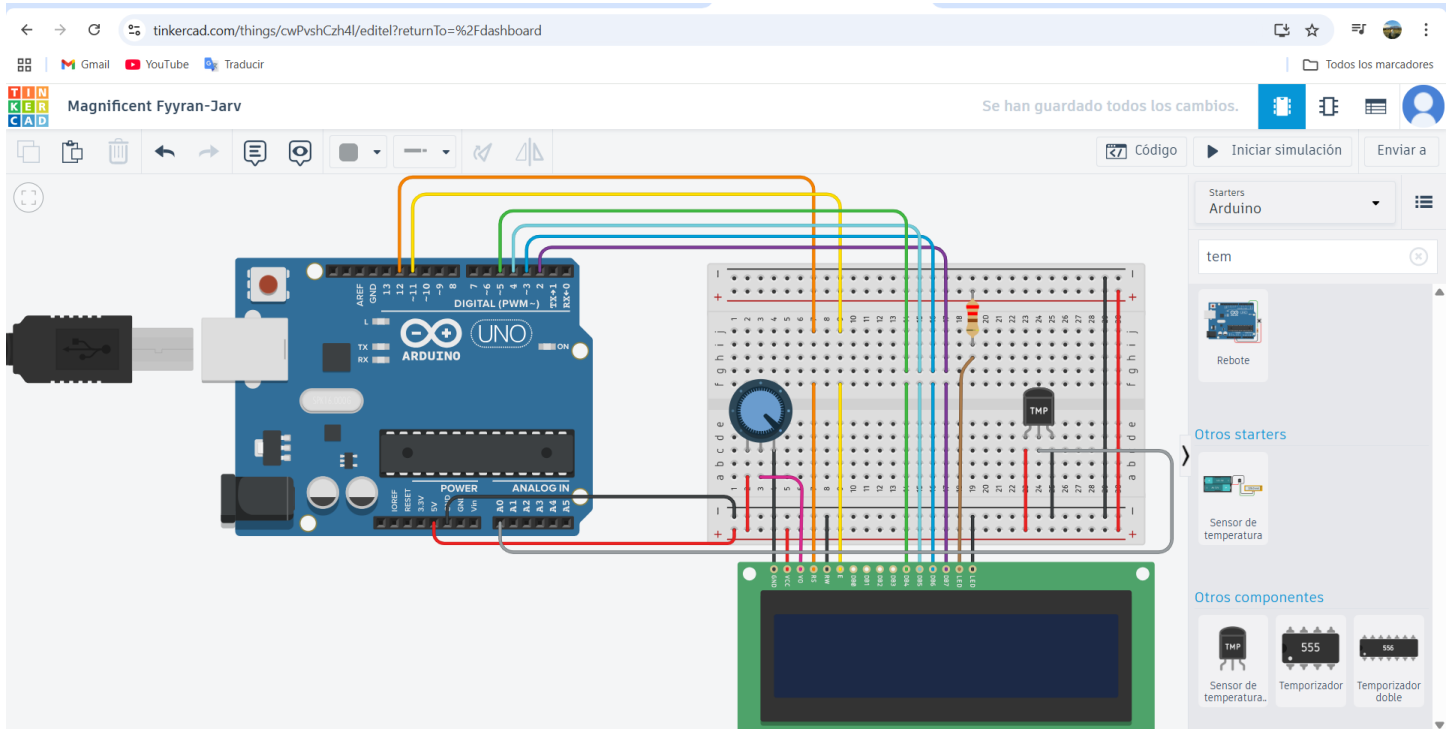
Para comenzar con el armado de nuestro circuito tomaremos como apoyo los siguientes componentes con la finalidad de agilizar la actividad el cual será un Arduino complementado con un LCD de 2 Cables, en donde podemos observar que al ejecutar nos manda un mensaje de “Hola Mundo”.



Posteriormente agregaremos un sensor de temperatura el cual brevemente conectaremos a voltaje y GND.

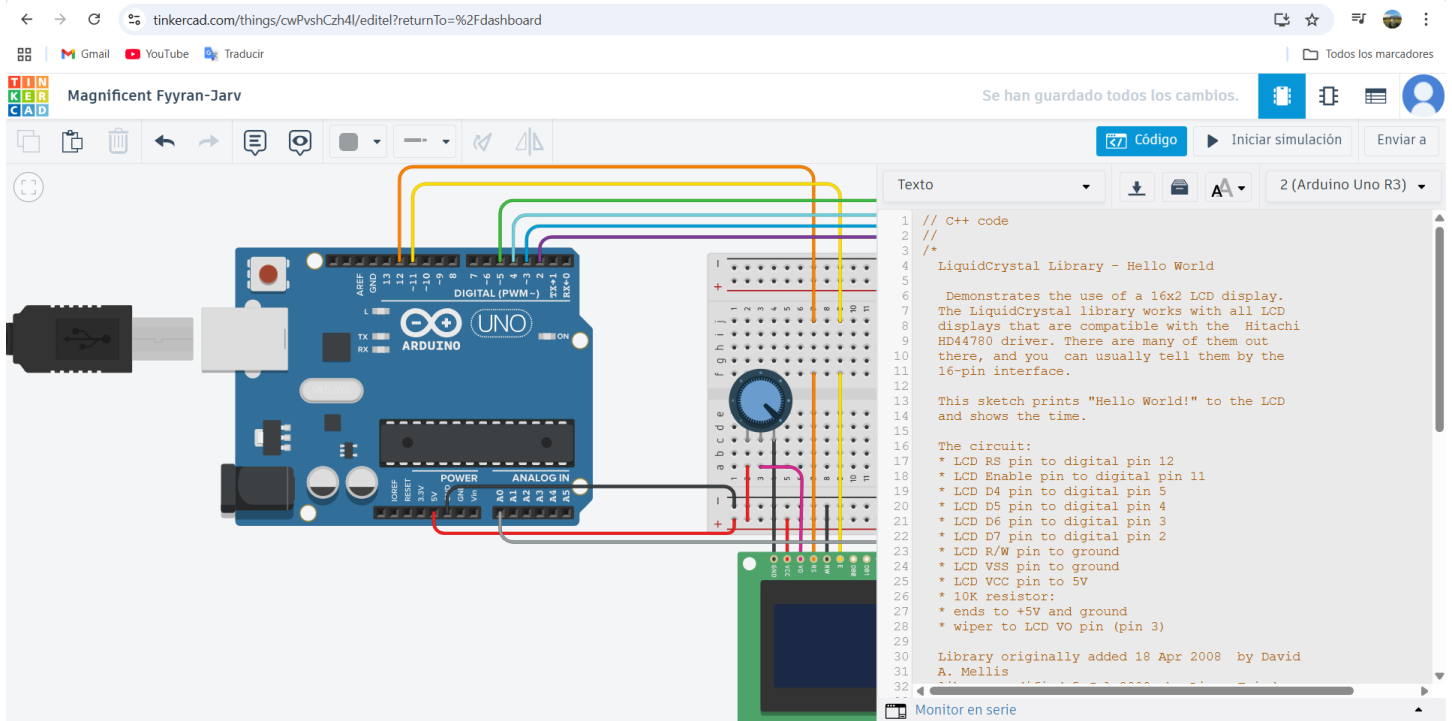


Continuaremos conectado al puerto analógico nuestro sensor en donde conectaremos el voltaje a nuestra placa Arduino, en este caso lo conectaremos al puerto A0.

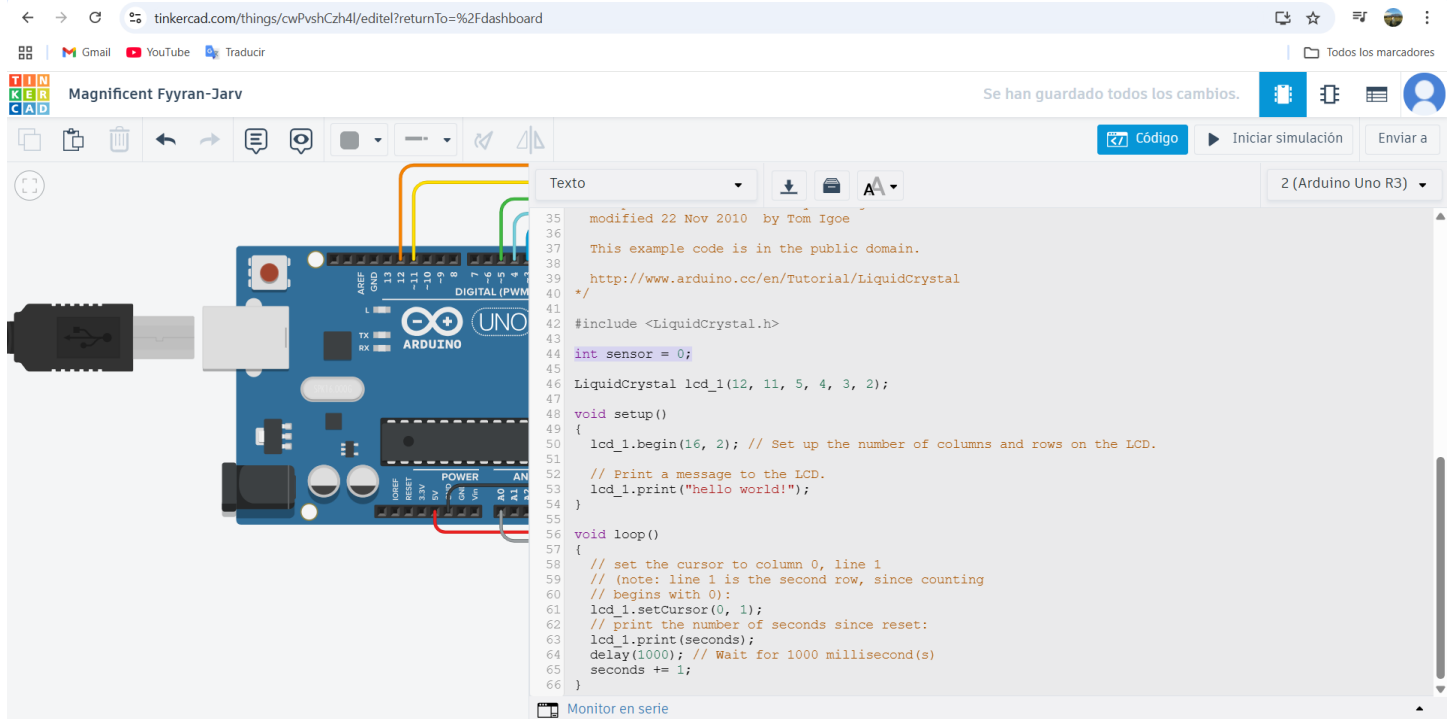


## Codificación

Continuaremos con la parte de la codificación del circuito el cual haremos uso de la librería LiquidCrystal que ya está previamente cargada en el código.



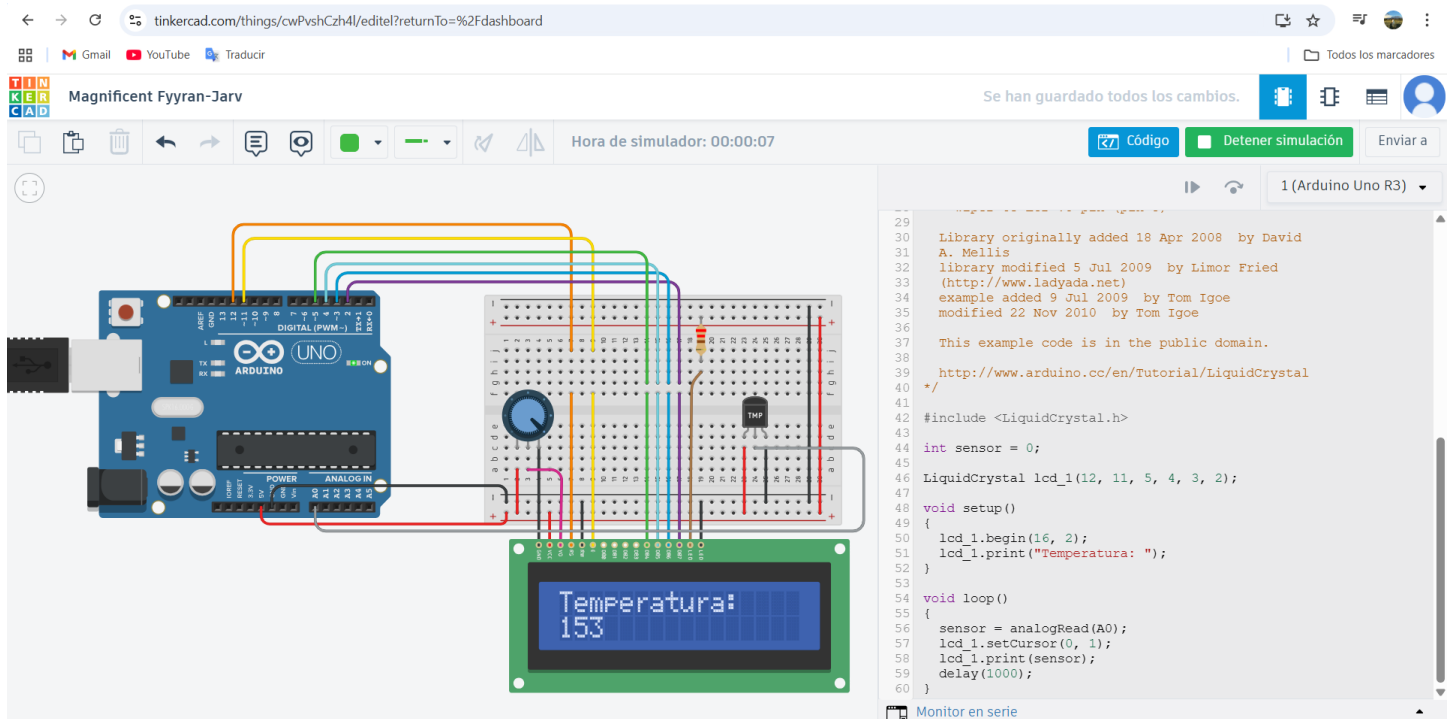
Declaramos la variable de tipo entero llamada sensor en donde esta nos servirá para guardar la señal analógica.



The screenshot shows the Tinkercad web interface with an Arduino Uno connected to an LCD screen. The code editor displays the following code:

```
35 modified 22 Nov 2010 by Tom Igoe
36 This example code is in the public domain.
37 http://www.arduino.cc/en/Tutorial/LiquidCrystal
38 */
39
40 #include <LiquidCrystal.h>
41
42 int sensor = 0;
43
44 LiquidCrystal lcd_1(12, 11, 5, 4, 3, 2);
45
46 void setup()
47 {
48   lcd_1.begin(16, 2); // Set up the number of columns and rows on the LCD.
49   // Print a message to the LCD.
50   lcd_1.print("hello world!");
51 }
52
53 void loop()
54 {
55   // set the cursor to column 0, line 1
56   // (note: line 1 is the second row, since counting
57   // begins with 0):
58   lcd_1.setCursor(0, 1);
59   // print the number of seconds since reset:
60   lcd_1.print(seconds);
61   delay(1000); // Wait for 1000 millisecond(s)
62   seconds += 1;
63 }
64
65 Monitor en serie
```

Continuamos con la asignación de nuestra variable sensor con la línea de código analogRead indicando el puerto al que esta conectado, en donde toda la señal arrojada será guardada en nuestra variable.



The screenshot shows the Tinkercad web interface with the same Arduino Uno setup, but now with a temperature sensor connected to the analog input. The LCD screen displays "Temperatura: 153". The code editor displays the following code:

```
29
30 Library originally added 18 Apr 2008 by David
31 A. Mellis
32 library modified 5 Jul 2009 by Limor Fried
33 (http://www.ladyada.net)
34 example added 9 Jul 2009 by Tom Igoe
35 modified 22 Nov 2010 by Tom Igoe
36
37 This example code is in the public domain.
38 http://www.arduino.cc/en/Tutorial/LiquidCrystal
39 */
40
41 #include <LiquidCrystal.h>
42
43 int sensor = 0;
44
45 LiquidCrystal lcd_1(12, 11, 5, 4, 3, 2);
46
47 void setup()
48 {
49   lcd_1.begin(16, 2);
50   lcd_1.print("Temperatura: ");
51 }
52
53 void loop()
54 {
55   sensor = analogRead(A0);
56   lcd_1.setCursor(0, 1);
57   lcd_1.print(sensor);
58   delay(1000);
59 }
60
61 Monitor en serie
```

tinkercad.com/things/cwPvshCzh4l/editel?returnTo=%2Fdashboard

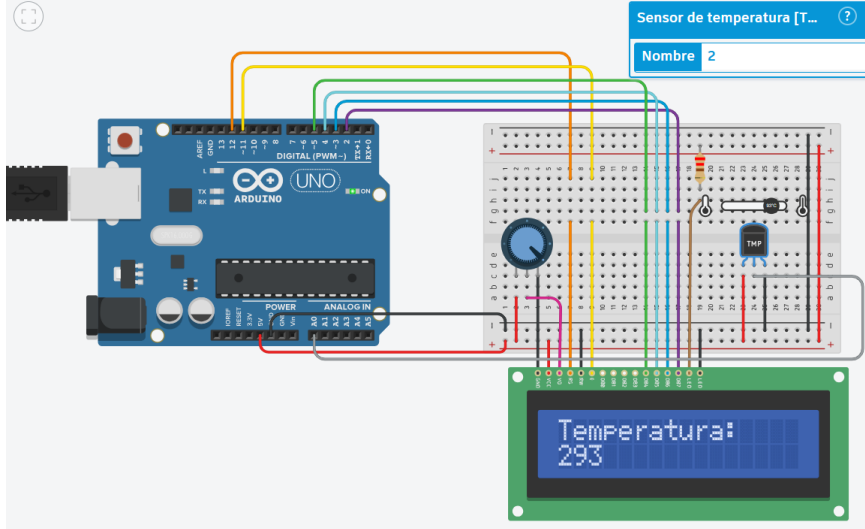
Magnificent Fyrran-Jarv

Se han guardado todos los cambios.

Hora de simulador: 00:00:08

Código Detener simulación Enviar a

1 (Arduino Uno R3)



Sensor de temperatura [T... ?]  
Nombre 2

```

29
30 Library originally added 18 Apr 2008 by David
31 A. Mellis
32 library modified 5 Jul 2009 by Limor Fried
33 (http://www.ladyada.net)
34 example added 9 Jul 2009 by Tom Igoe
35 modified 22 Nov 2010 by Tom Igoe
36
37 This example code is in the public domain.
38
39 http://www.arduino.cc/en/Tutorial/LiquidCrystal
40 */
41
42 #include <LiquidCrystal.h>
43
44 int sensor = 0;
45
46 LiquidCrystal lcd_1(12, 11, 5, 4, 3, 2);
47
48 void setup()
49 {
50   lcd_1.begin(16, 2);
51   lcd_1.print("Temperatura: ");
52 }
53
54 void loop()
55 {
56   sensor = analogRead(A0);
57   lcd_1.setCursor(0, 1);
58   lcd_1.print(sensor);
59   delay(1000);
60 }

```

Monitor en serie

Una vez realizado lo anterior realizaremos la conversión a grados centígrados, para esto declararemos una variable tipo flotante para obtener la temperatura en grados centígrados en donde realizaremos una formula para poder calcular grados centígrados.

tinkercad.com/things/cwPvshCzh4l/editel?returnTo=%2Fdashboard

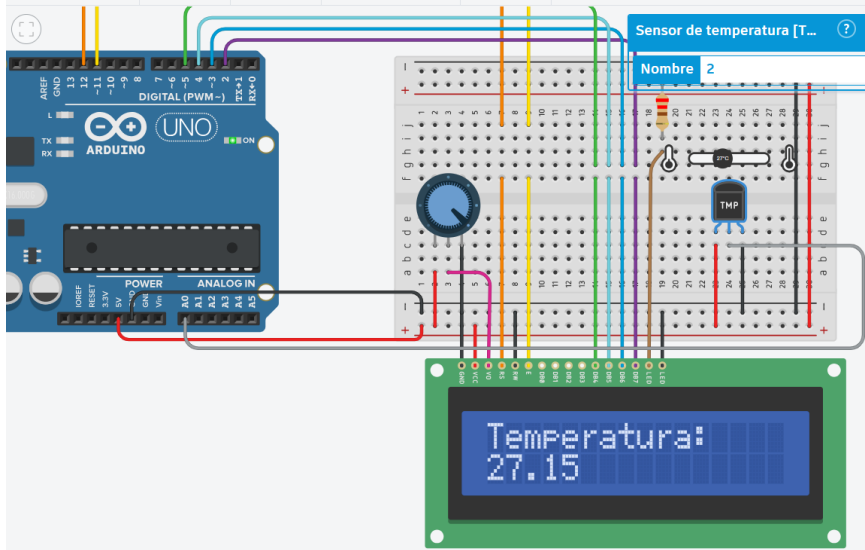
Magnificent Fyrran-Jarv

Se han guardado todos los cambios.

Hora de simulador: 00:00:37

Código Detener simulación Enviar a

1 (Arduino Uno R3)



Sensor de temperatura [T... ?]  
Nombre 2

```

30 Library originally added 18 Apr 2008 by David
31 A. Mellis
32 library modified 5 Jul 2009 by Limor Fried
33 (http://www.ladyada.net)
34 example added 9 Jul 2009 by Tom Igoe
35 modified 22 Nov 2010 by Tom Igoe
36
37 This example code is in the public domain.
38
39 http://www.arduino.cc/en/Tutorial/LiquidCrystal
40 */
41
42 #include <LiquidCrystal.h>
43
44 int sensor = 0;
45
46 LiquidCrystal lcd_1(12, 11, 5, 4, 3, 2);
47
48 void setup()
49 {
50   lcd_1.begin(16, 2);
51   lcd_1.print("Temperatura: ");
52 }
53
54 void loop()
55 {
56   sensor = analogRead(A0);
57   lcd_1.setCursor(0, 1);
58   float tempC = (((float) sensor / 1024)*5-0.5) *100;
59   lcd_1.print(tempC);
60   delay(1000);
61 }

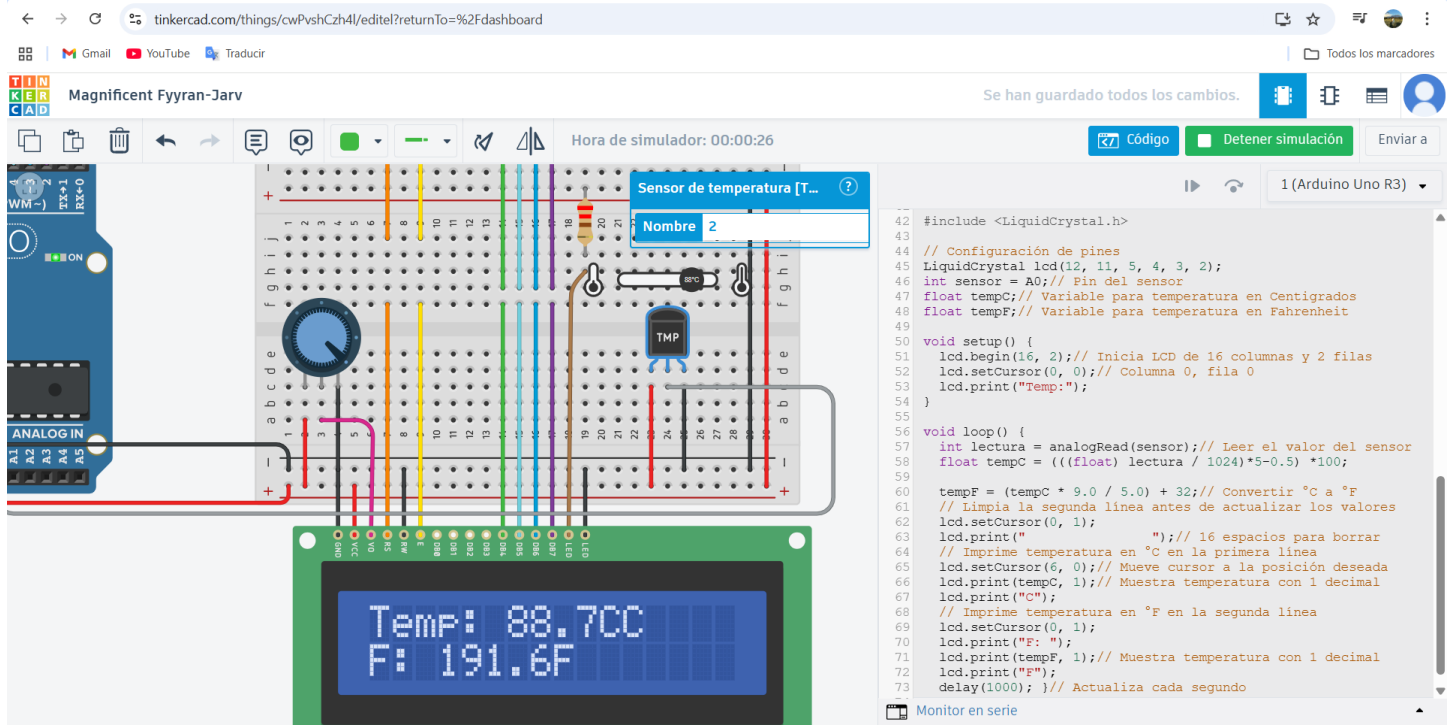
```

Monitor en serie



Posteriormente se realizó una modificación en el código con el fin de poder obtener tanto la temperatura en grados centígrados y Fahrenheit.

Se agregan distintas temperaturas ejecutando nuestro circuito.

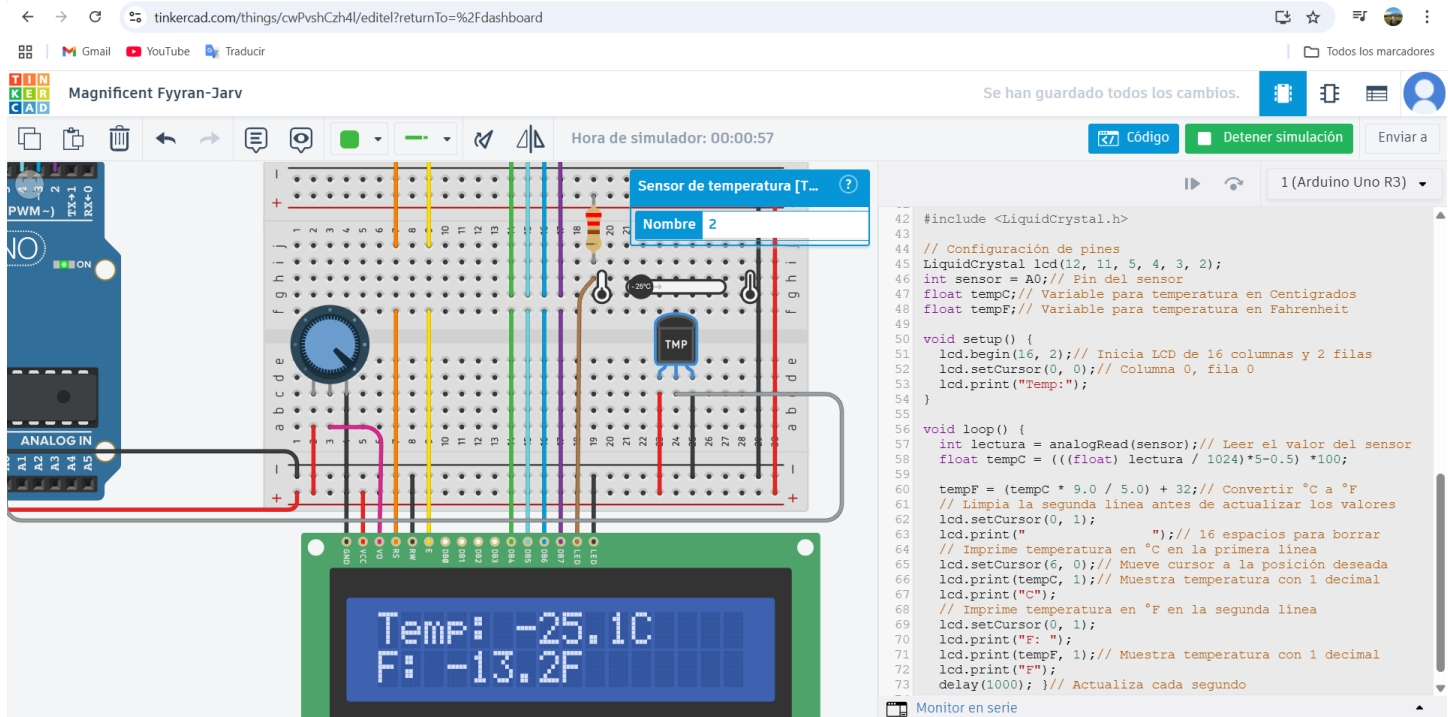


The screenshot shows the Tinkercad web interface with a breadboard circuit and its corresponding code. The circuit includes an Arduino Uno R3, a temperature sensor (TMP), and an LCD display. The LCD shows 'Temp: 88.7°C' and 'F: 191.6°F'. The code is in C++ and includes comments in Spanish.

```
#include <LiquidCrystal.h>
// Configuración de pines
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int sensor = A0; // Pin del sensor
float tempC; // Variable para temperatura en Centígrados
float tempF; // Variable para temperatura en Fahrenheit

void setup() {
  lcd.begin(16, 2); // Inicia LCD de 16 columnas y 2 filas
  lcd.setCursor(0, 0); // Columna 0, fila 0
  lcd.print("Temp:");
}

void loop() {
  int lectura = analogRead(sensor); // Leer el valor del sensor
  float tempC = ((float) lectura / 1024) * 5 - 0.5; // Convertir a °C
  tempF = (tempC * 9.0 / 5.0) + 32; // Convertir °C a °F
  // Limpia la segunda línea antes de actualizar los valores
  lcd.setCursor(0, 1);
  // Imprime temperatura en °C en la primera línea
  lcd.setCursor(6, 0); // Mueve cursor a la posición deseada
  lcd.print(tempC, 1); // Muestra temperatura con 1 decimal
  lcd.print("°C");
  // Imprime temperatura en °F en la segunda línea
  lcd.setCursor(0, 1);
  lcd.print("°F: ");
  lcd.print(tempF, 1); // Muestra temperatura con 1 decimal
  lcd.print("°F");
  delay(1000); // Actualiza cada segundo
```



The screenshot shows the Tinkercad web interface with the same breadboard circuit as above, but with different temperature readings on the LCD: 'Temp: -25.1°C' and 'F: -13.2°F'. The code is also shown, with some differences in the conversion formulas.

```
#include <LiquidCrystal.h>
// Configuración de pines
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int sensor = A0; // Pin del sensor
float tempC; // Variable para temperatura en Centígrados
float tempF; // Variable para temperatura en Fahrenheit

void setup() {
  lcd.begin(16, 2); // Inicia LCD de 16 columnas y 2 filas
  lcd.setCursor(0, 0); // Columna 0, fila 0
  lcd.print("Temp:");
}

void loop() {
  int lectura = analogRead(sensor); // Leer el valor del sensor
  float tempC = ((float) lectura / 1024) * 5 - 0.5; // Convertir a °C
  tempF = (tempC * 9.0 / 5.0) + 32; // Convertir °C a °F
  // Limpia la segunda línea antes de actualizar los valores
  lcd.setCursor(0, 1);
  // Imprime temperatura en °C en la primera línea
  lcd.setCursor(6, 0); // Mueve cursor a la posición deseada
  lcd.print(tempC, 1); // Muestra temperatura con 1 decimal
  lcd.print("°C");
  // Imprime temperatura en °F en la segunda línea
  lcd.setCursor(0, 1);
  lcd.print("°F: ");
  lcd.print(tempF, 1); // Muestra temperatura con 1 decimal
  lcd.print("°F");
  delay(1000); // Actualiza cada segundo
```

## Enlaces.

<https://www.tinkercad.com/things/cwPvshCzh4I-magnificent-fyran-jarv>

<https://github.com/monroy2321eduardo-sudo/Internet-De-La-Cosas->

## Conclusión

En esta actividad aprendí a usar un sensor de temperatura junto con una pantalla LCD y una placa Arduino. Con el trabajo en Tinkercad pude ver cómo se conectan los componentes y cómo el programa hace que todo funcione. El sensor detecta la temperatura del ambiente y el Arduino manda ese valor a la pantalla, donde se muestra en grados Celsius. Me pareció muy interesante ver cómo algo que parece complicado se puede entender paso a paso.

También aprendí que la función LiquidCrystal es muy importante, porque sirve para que la pantalla pueda mostrar letras o números. Además, supe que se deben usar variables tipo float para poder guardar números con decimales, ya que la temperatura no siempre es un número exacto. Usando las funciones MAP y analogRead, se obtiene el valor que mide el sensor, y después al dividirlo entre 100 se muestra correctamente en la pantalla.

Gracias a esta práctica entendí mejor cómo los sensores y las pantallas trabajan juntos para mostrar información. Esto se usa en muchas cosas de la vida diaria, como los termómetros digitales, los refrigeradores o los sistemas inteligentes que controlan la temperatura.

En resumen, esta actividad me ayudó a entender cómo el Arduino puede recibir datos del entorno y mostrarlos de una forma que todos podamos ver. También me sirvió para mejorar mis habilidades con los circuitos y la programación básica. Fue una experiencia muy útil y me hizo ver cómo la tecnología del Internet de las Cosas se puede aplicar en cosas simples pero importantes.