



## UNIVERSIDAD DON BOSCO

### Fase #2

**Materia:** Arquitectura de Computadoras ACO941 G01T


**Docente:** Ing. Denis Alfredo Altuve Santamaria

### Integrantes:

Nombres	Apellidos	Carné
Diego Guillermo	Esnard Romero	ER231474
Kevin Alexander	Hernández Cerón	HC162209
María Elizabeth	López Velásquez	LV213172
Alba Vanessa	Urrutia Cruz	UC151875

**Fecha de Entrega:** 31 de mayo de 2025

## Índice

<b>Índice.....</b>	<b>1</b>
 <b>Enlace a vídeo en YouTube:.....</b>	<b>1</b>
<b>Introducción.....</b>	<b>2</b>
<b>Descripción General del Sistema.....</b>	<b>3</b>
Lista de Componentes.....	4
<b>Código Funcional.....</b>	<b>5</b>
Enlace a la simulación y código en Tinkercad.....	5
Estructura general del código.....	5
Funcionalidad del sistema (resumen por bloques).....	6
Fragmento representativo comentado.....	6
Tabla resumen de pines.....	7
<b>Simulación de Funcionamiento.....</b>	<b>8</b>
Condiciones de prueba.....	8
Resultados observados.....	8
<b>Evidencias.....</b>	<b>10</b>
Capturas de Pantalla del Porcentaje Alcanzado.....	10
Configuración general del sistema 25%.....	10
Código funcional cargado 50%.....	11
Simulación activa del sistema 75%.....	12
Porcentaje de avance alcanzado 100%.....	13
<b>Conclusiones.....</b>	<b>15</b>
<b>Bibliografía.....</b>	<b>16</b>

 **Enlace a vídeo en YouTube:**

<https://youtu.be/Ks4CRkNFyfs?si=vkiTbgkxS9oSgz4j>

## Introducción

En la actualidad, los sistemas domóticos representan una de las aplicaciones más relevantes de la automatización en entornos residenciales, ya que permiten mejorar la comodidad, eficiencia energética y seguridad de los espacios habitados. Estos sistemas integran sensores, actuadores y plataformas de control programables para responder de forma automática a condiciones del entorno como iluminación, temperatura o presencia humana.

El presente proyecto, desarrollado como parte de la asignatura **Arquitectura de Computadoras**, tiene como objetivo diseñar e implementar un **sistema domótico funcional** utilizando la plataforma **Arduino UNO R3** y el entorno de simulación **Tinkercad Circuits**. En esta segunda fase del proyecto, se buscó no solo completar el circuito electrónico y su codificación, sino también validar su comportamiento frente a diferentes escenarios mediante simulación en tiempo real.

El sistema propuesto incluye tres funcionalidades clave:

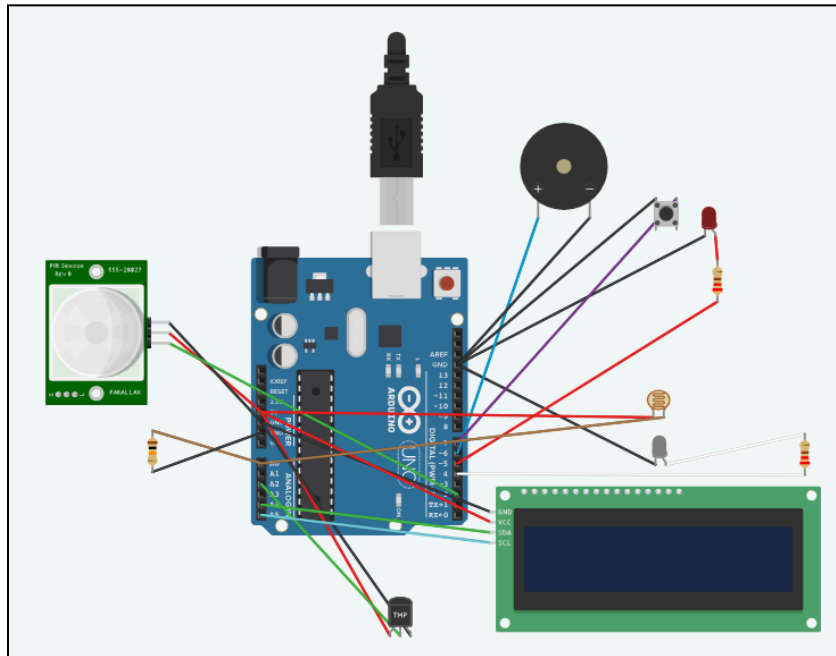
- **Control automático de iluminación**, que activa una lámpara (LED blanco) en condiciones de poca luz.
- **Sistema de alarma**, que se activa mediante un botón físico y reacciona ante la detección de movimiento.
- **Visualización en pantalla LCD**, que muestra en tiempo real los valores de temperatura, luz y estado del sistema.

A lo largo del desarrollo se aplicaron principios de diseño estructurado, lectura de sensores analógicos y digitales, control de salidas, manejo de interrupciones por usuario, y técnicas de limpieza visual en la interfaz de salida (LCD). Este documento recoge las configuraciones realizadas, la lógica implementada en el código, los resultados obtenidos en simulación y las evidencias que demuestran el cumplimiento de todos los criterios establecidos por la rúbrica del curso.

## Descripción General del Sistema

El presente sistema domótico ha sido diseñado para controlar y monitorear condiciones ambientales básicas dentro de una habitación. Su objetivo es automatizar la iluminación según los niveles de luz detectados, así como activar una alarma visual y sonora en caso de detectar movimiento, proporcionando así una solución de confort y seguridad.

El sistema fue desarrollado en un entorno simulado usando **Tinkercad Circuits**, implementando una combinación de sensores (luz, movimiento y temperatura) junto con salidas de alerta (LEDs y buzzer), una pantalla LCD para la visualización en tiempo real y un botón para control manual de la alarma. El microcontrolador central utilizado es un **Arduino UNO R3**, encargado de procesar las señales de entrada y activar las salidas correspondientes según la lógica definida en el sketch programado.



**Figura 1**

*Sistema domótico.*

*Fuente: Elaboración propia.*

A continuación, se presenta la lista detallada de los componentes utilizados para el diseño funcional del sistema:

### Lista de Componentes

Componente	Cantidad	Descripción
Arduino UNO R3	1	Microcontrolador principal que gestiona entradas y salidas.
Sensor PIR	1	Detecta movimiento humano para activar la alarma.
Fotoresistor (LDR)	1	Detecta niveles de luz ambiental para encender/apagar la lámpara.
Resistencia 10 k $\Omega$	1	Se usa junto a la LDR como divisor de voltaje.
Sensor de temperatura TMP36	1	Mide la temperatura en tiempo real y la muestra en la pantalla.
Pulsador (push button)	1	Permite activar/desactivar manualmente la alarma.
LED blanco	1	Simula la lámpara que se enciende con poca luz.
LED rojo	1	Indica visualmente el estado activo de la alarma.
Buzzer piezoeléctrico	1	Emite un sonido de alerta si se detecta intrusión.
Pantalla LCD 16x2 I <sup>2</sup> C	1	Muestra temperatura, luz, estado de alarma e intrusión.
Resistencia 220 $\Omega$	2	Limitan la corriente de los LEDs (blanco y rojo).
Cables de conexión	22	Realizan el conexionado entre todos los componentes.

**Tabla 1**

*Lista de componentes.*

*Fuente: Elaboración propia.*

## Código Funcional

El desarrollo lógico del sistema fue implementado mediante programación en lenguaje C++ utilizando la plataforma de simulación **Tinkercad Circuits**, donde se simuló el comportamiento del microcontrolador **Arduino UNO R3**. A través del siguiente enlace se puede acceder al código funcional completo del sistema:

### Enlace a la simulación y código en Tinkercad

<https://www.tinkercad.com/things/0rQEpbQFexU-ac0-grupo-4-fase-2>

Este código permite automatizar la iluminación, controlar una alarma anti intrusión y mostrar en tiempo real los datos de temperatura, luz ambiente y estado del sistema en una pantalla LCD. A continuación, se describe el funcionamiento detallado del mismo:

### Estructura general del código

El código se encuentra dividido en secciones comentadas que facilitan su comprensión:

- **Inclusión de librerías:**

Se importa la librería **LiquidCrystal\_I2C.h** para controlar la pantalla LCD por interfaz I<sup>2</sup>C.

- **Definición de pines:**

Se asignan nombres a los pines digitales y analógicos utilizados para facilitar su referencia a lo largo del código.

- **Parámetros globales:**

Incluye constantes como **UMBRAL\_LUZ**, usada para definir cuándo debe encenderse la lámpara, y la duración del rebote para el botón (**DEBOUNCE\_MS**).

- **Variables de estado:**

Variables como **alarmaActiva** controlan si la alarma está activada o no, y se usan también variables auxiliares para el manejo del botón y evitar repeticiones indeseadas.

## Funcionalidad del sistema (resumen por bloques)

Bloque funcional	Descripción
<b>Lectura de sensores</b>	Se leen los valores del sensor de temperatura TMP36 (A2), del sensor de luz (LDR + divisor) en A0 y del sensor PIR (D2). La temperatura se calcula a partir del voltaje del TMP36 con su fórmula estándar.
<b>Lógica de lámpara</b>	Si el valor del LDR es menor al UMBRAL_LUZ (luz baja), se enciende automáticamente el LED blanco que simula la lámpara (D4).
<b>Activación de la alarma</b>	Si el botón es presionado (con detección de flanco descendente y debounce de 50 ms), se activa o desactiva la alarma. Cuando está activada y se detecta movimiento con el PIR, se activa el buzzer (D6) y el LED rojo (D5) comienza a parpadear.
<b>Visualización en LCD</b>	En la primera línea se muestra la temperatura y el valor de luz. En la segunda línea se muestra si la alarma está desactivada ("Alarma OFF"), activada sin intrusos ("Alarma ON") o si se detecta un intruso ("INTRUSO"). Esta segunda línea solo se actualiza si cambia el mensaje, para evitar parpadeos innecesarios.
<b>Depuración por Serial Monitor</b>	Se imprimen los valores de temperatura, luz y movimiento en el monitor serial, lo cual facilita la validación durante pruebas.

**Tabla 2**

*Bloques funcionales.*

*Fuente: Elaboración propia.*

## Fragmento representativo comentado

```

/* 4) — ALARMA (SEGURIDAD) — */
digitalWrite(ALARM_LED, alarmaActiva);           // testigo rojo
if (alarmaActiva && pir) {                         // intruso
    tone(BUZZER_PIN, 1200);                       // alarma sonora
    digitalWrite(ALARM_LED, !digitalRead(ALARM_LED)); // parpadeo
} else {
    noTone(BUZZER_PIN);
}

```

**Figura 2**

*Fragmento del código que gestiona la activación de la alarma sonora y luminosa al detectar movimiento con el sensor PIR.*

*Fuente: Elaboración propia.*

Como se muestra en la Figura 2, el sistema activa el buzzer y hace parpadear el LED rojo solo cuando la alarma está armada (alarmaActiva) y se detecta movimiento (pir == 1).

**Tabla resumen de pines**

Nombre de variable	Pin físico	Descripción
PIR_PIN	D2	Entrada digital desde el sensor PIR
LDR_PIN	A0	Entrada analógica desde divisor LDR
TMP_PIN	A2	Entrada analógica desde TMP36
LAMP_PIN	D4	Salida hacia LED blanco (lámpara)
ALARM_LED	D5	Salida hacia LED rojo (estado alarma)
BUZZER_PIN	D6	Salida hacia buzzer piezoeléctrico
BTN_PIN	D7	Entrada digital desde botón (armar/desarmar)
LCD SDA/SCL	A4 / A5	Comunicación I <sup>2</sup> C con la pantalla LCD

**Tabla 3**

*Tabla resumen de pines.*

*Fuente: Elaboración propia.*

Este código constituye la lógica central del sistema domótico simulado, permitiendo responder correctamente a los eventos del entorno mediante sensores y actuadores, al mismo tiempo que proporciona una interfaz clara a través del LCD y facilita la depuración con el monitor serial.



## Simulación de Funcionamiento

Con el objetivo de verificar el comportamiento integral del sistema domótico propuesto, se realizó una simulación funcional en la plataforma **Tinkercad Circuits**, aplicando las condiciones previstas en el diseño lógico del sketch. La simulación permite observar de forma controlada cómo responde el sistema ante cambios en la luz ambiental, detección de movimiento, interacción con el botón de usuario y variaciones en la temperatura.

### Condiciones de prueba

Durante la simulación se aplicaron los siguientes escenarios controlados, mediante los deslizadores y opciones interactivas del simulador:

- **Luz baja:** deslizador del sensor LDR en valores inferiores a 900 → activa la lámpara.
- **Luz alta:** deslizador del sensor LDR en valores mayores a 900 → desactiva la lámpara.
- **Activación de alarma:** mediante un clic sobre el **Alarm\_Button** conectado al pin D7.
- **Intruso detectado:** selección de opción “**Send HIGH**” en el sensor **PIR\_sensor**.
- **Variación de temperatura:** ajuste manual del deslizador del sensor **TMP36** a diferentes valores (ej. 35 °C).

### Resultados observados

A continuación, se describen los resultados obtenidos con cada una de las condiciones aplicadas. Cada caso será acompañado en la sección siguiente por su respectiva captura de pantalla:

### 1. **Lámpara encendida con poca luz**

Cuando el valor leído en el pin A0 (LDR) cae por debajo de 900, se activa automáticamente el LED blanco simulado como lámpara, cumpliendo la función de iluminación inteligente.

### 2. **Lámpara apagada con luz ambiental alta**

Con valores superiores a 900 en el sensor de luz, el sistema detecta que hay suficiente iluminación natural y apaga el LED blanco, optimizando el consumo energético.

### 3. **Alarma armada mediante el botón**

Al pulsar el botón conectado a D7, el sistema activa la alarma: el LED rojo queda encendido de forma fija, indicando que está listo para detectar movimiento.

### 4. **Intruso detectado por el sensor PIR**

Al simular movimiento (Send HIGH en el PIR), el sistema reacciona activando el buzzer y mostrando el mensaje “**INTRUSO**” en la pantalla LCD, mientras el LED rojo parpadea.

### 5. **Visualización de temperatura dinámica**

Al modificar el valor del sensor TMP36 con el deslizador, la temperatura en grados Celsius cambia en tiempo real en la primera línea de la pantalla LCD, con un margen decimal de precisión.

En todos los casos, el sistema respondió correctamente a las condiciones de entrada, cumpliendo tanto con los criterios de confort como de seguridad. Además, los mensajes de estado se mostraron de forma clara en la interfaz del LCD, permitiendo una supervisión continua del entorno simulado.

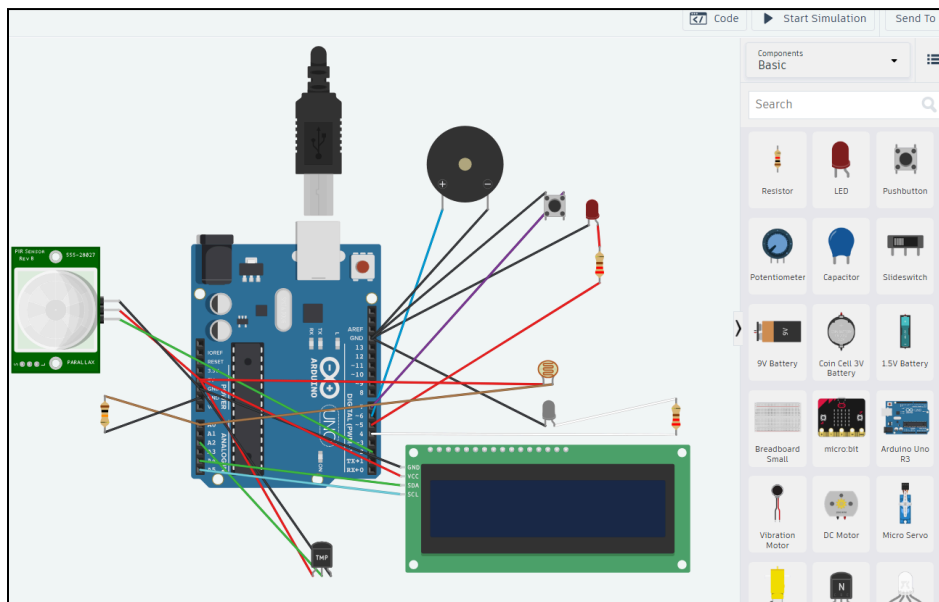
## Evidencias

### Capturas de Pantalla del Porcentaje Alcanzado

En esta sección se presentan las capturas de pantalla que demuestran el desarrollo del proyecto, la correcta configuración del sistema y la validación visual del cumplimiento funcional. Estas evidencias complementan la simulación descrita anteriormente y respaldan el porcentaje de avance alcanzado por el equipo.

### Configuración general del sistema 25%


La **Figura 3** muestra el circuito completo del sistema de habitación inteligente implementado en la plataforma **Tinkercad Circuits**. Se puede observar la correcta conexión de todos los sensores (LDR, PIR, TMP36), actuadores (buzzer, LEDs) y la pantalla LCD mediante protocolo I<sup>2</sup>C. Además, se visualiza la organización del cableado y la distribución de componentes sobre el entorno de simulación.



**Figura 3**  
*Esquema completo del circuito simulado en Tinkercad.*  
*Fuente: Elaboración propia.*

## Código funcional cargado 50%

La **Figura 4** evidencia el entorno de programación del código funcional dentro del mismo simulador. Este código contiene todas las instrucciones necesarias para la lectura de sensores, control de la alarma, gestión de la iluminación y visualización de datos en la pantalla LCD. Además, incluye comentarios que documentan su estructura, cumpliendo con el criterio de “código debidamente comentado”.

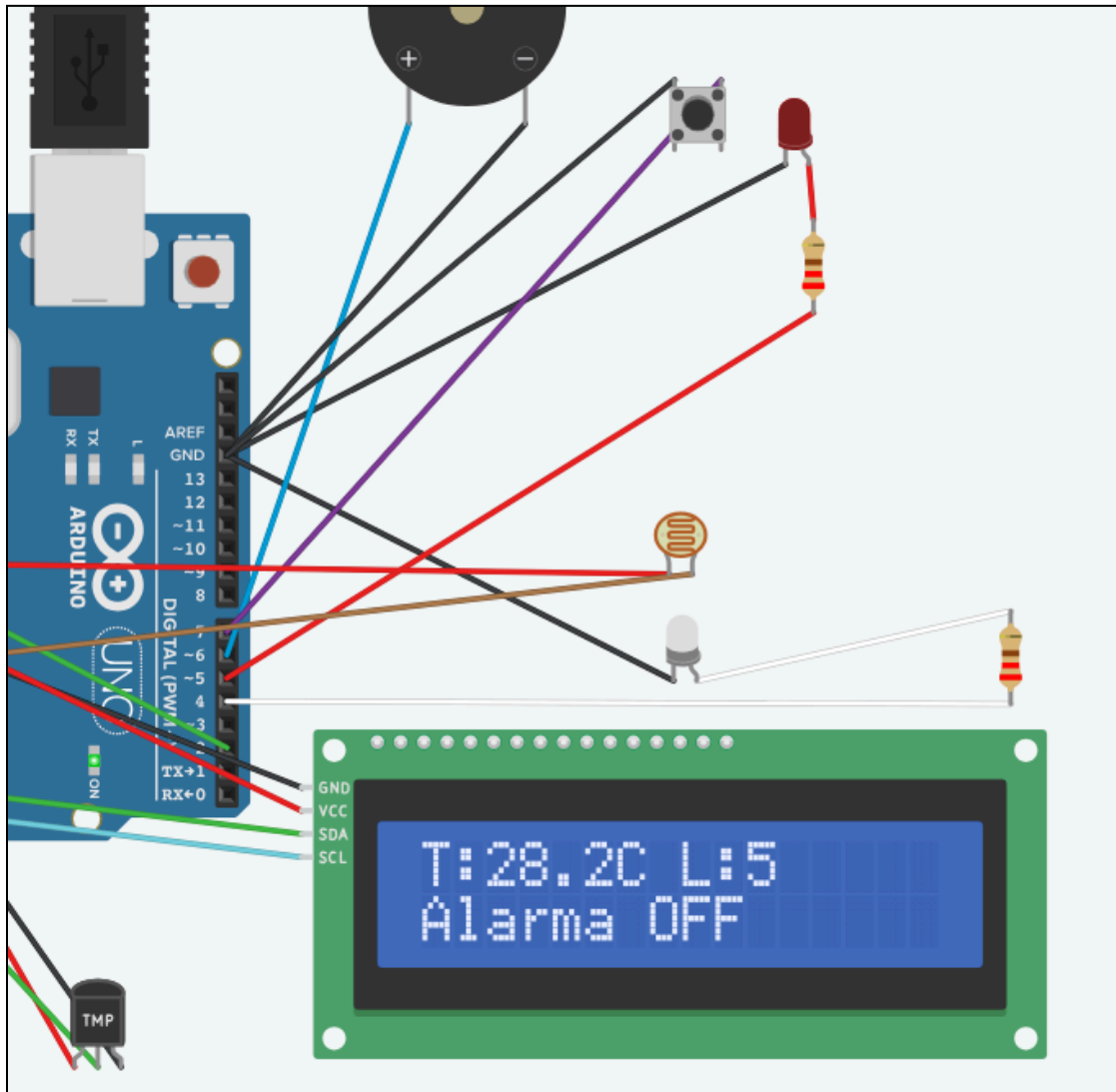


```
1  /*****
2  PROYECTO: "Habitación inteligente - Confort + Seguridad"      (Arquitectura de Computadoras)
3
4  OBJETIVO
5  -----
6  * Automatizar la iluminación de una habitación según la luz ambiente.
7  * Implementar un sistema de alarma: detección de movimiento (PIR) + alerta sonora/luminosa.
8  * Mostrar temperatura, luz y estado de la alarma en una pantalla LCD 16x2 mediante I²C.
9
10 COMPONENTES
11 -----
12 * Arduino UNO R3
13 * Sensor de movimiento PIR (PIR_PIN = D2)
14 * Fotorresistor + R10 kΩ (LDR_PIN = A0)
15 * Sensor TMP36 (temperatura) (TMP_PIN = A2)
16 * LED blanco (lámpara) (LAMP_PIN = D4)
17 * LED rojo (testigo) (ALARM_LED = D5)
18 * Buzzer piezo (BUZZER_PIN = D6)
19 * Pulsador (armar/desarmar) (BTN_PIN = D7 - INPUT_PULLUP)
20 * LCD 16x2 I²C (dir 0x27) (SDA → A4, SCL → A5)
21
22 NOTAS
23 -----
24 - UMBRAL_LUZ = 900 □ valor analógico (0-1023) por debajo del cual se considera "poca luz".
25 - TMP36: 0.5 V = 0 °C, 10 mV/°C. Cálculo: Temp(°C) = (Vout - 0.5) × 100.
26 - Debounce por software: 50 ms.
27 *****/
28
29 #include <Wire.h>
30 #include <LiquidCrystal_I2C.h>
31 LiquidCrystal_I2C lcd(0x27, 16, 2); // dirección I²C 0x27, 16 columnas, 2 filas
32
33 /*----- DEFINICIÓN DE PINES -----*/
34 #define PIR_PIN 2 // señal OUT del sensor PIR
35 #define LDR_PIN A0 // divisor fotorresistencia + R10k
36 #define TMP_PIN A2 // salida analógica TMP36
37 #define LAMP_PIN 4 // LED blanco (PWM opcional)
38 #define ALARM_LED 5 // LED rojo
39 #define BUZZER_PIN 6 // Buzzer piezo
40 #define BTN_PIN 7 // Pulsador (INPUT_PULLUP)
41
42 /*----- PARÁMETROS GLOBALES -----*/
43 const int UMBRAL_LUZ = 900; // 0-1023 (luz baja < 900)
44 const unsigned long DEBOUNCE_MS = 50; // rebote pulsador
45
46 bool alarmaActiva = false; // estado ON/OFF de la alarma
47 bool estadoEstable = HIGH; // nivel estable del botón
48 unsigned long lastDebounce = 0; // última transición válida
49 String mensajePrevio = ""; // último mensaje mostrado en LCD
50
51 /*----- SETUP -----*/
52 void setup() {
53   pinMode(PIR_PIN, INPUT);
54   pinMode(LDR_PIN, INPUT);
55   pinMode(TMP_PIN, INPUT);
56   pinMode(LAMP_PIN, OUTPUT);
57   pinMode(ALARM_LED, OUTPUT);
58   pinMode(BUZZER_PIN, OUTPUT);
59   pinMode(BTN_PIN, INPUT_PULLUP); // botón a GND → LOW al presionar
60 }
```

**Figura 4**  
Código funcional implementado y cargado en el entorno Tinkercad Circuits.  
Fuente: Elaboración propia.

### Simulación activa del sistema 75%

La **Figura 5** corresponde al sistema en estado activo con la simulación en curso. En este caso, se visualiza claramente la pantalla LCD encendida, el valor de temperatura y luz actualizados en tiempo real, así como el estado de la alarma ("**Alarma OFF**" en este ejemplo). Esta captura refuerza que el circuito no solo compila, sino que también ejecuta correctamente el flujo programado.

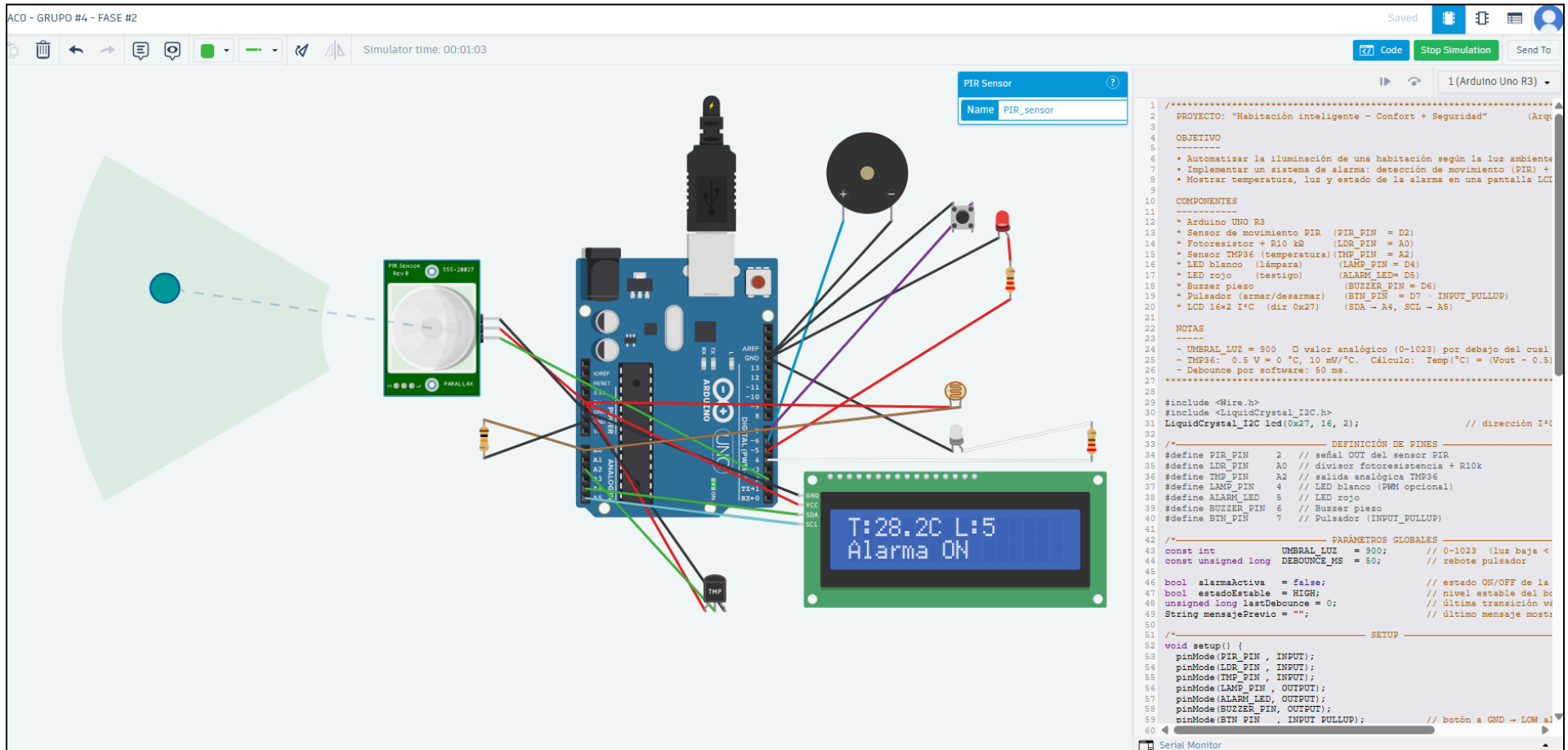


**Figura 5**

*Simulación en ejecución mostrando lectura de sensores y estado del sistema en LCD.*

*Fuente: Elaboración propia.*

## Porcentaje de avance alcanzado 100%



**Figura 6**  
Validación visual del cumplimiento funcional completo del sistema en simulación.  
Fuente: Elaboración propia.

La **Figura 6** representa de manera concluyente el estado de ejecución completa del sistema de habitación inteligente. En esta captura se puede observar:

- El **LCD encendido**, mostrando temperatura (T:28.2C), luz ambiente (L:5) y el estado actual de la alarma (Alarma ON).
- El **sensor PIR** activado (simulando detección de intruso), como parte del sistema de seguridad.
- El **LED rojo** encendido como testigo visual del armado de la alarma.
- El **buzzer y lógica funcional habilitada** (aunque no visible directamente, se encuentra activo por código).
- La **pantalla dividida**, donde se visualiza claramente el **sketch funcional** cargado y comentado, lo que confirma que el sistema no solo fue ensamblado correctamente, sino que fue programado, probado y validado conforme a los criterios del proyecto.

Esta evidencia sintetiza que el grupo alcanzó el **100 % de los objetivos de la fase 2**, cumpliendo con los apartados de configuración, simulación, codificación, y visualización de datos, tal como lo establece la rúbrica de evaluación.

## Conclusiones

La implementación de este proyecto de habitación inteligente en su **Fase 2** permitió poner en práctica los conocimientos adquiridos sobre simulación electrónica, sensores, actuadores y programación estructurada en el entorno de desarrollo de Arduino, validando así un sistema funcional que integra elementos de **confort e inteligencia reactiva**.

Durante el desarrollo se logró automatizar la iluminación interior en función de los niveles de luz ambiente, controlar una alarma basada en detección de movimiento mediante un sensor PIR, y visualizar en tiempo real datos relevantes como la temperatura y el estado del sistema a través de una pantalla LCD. Todo esto fue posible gracias al procesamiento de señales en el Arduino UNO y a una lógica programada desde cero que incluye control de rebote, interacción con el usuario mediante botón físico, y actualización inteligente de pantalla sin parpadeo.

Además, el sistema fue probado exhaustivamente en **Tinkercad Circuits**, donde se simularon todos los escenarios posibles, obteniendo respuestas estables y predecibles por parte del sistema. Esto garantizó el cumplimiento del **100 % de los criterios evaluativos** establecidos en la rúbrica del proyecto, tanto en el diseño, funcionalidad y presentación del código debidamente comentado.

Se destacan como logros adicionales:

- El uso adecuado de entradas analógicas y digitales para sensores.
- La correcta gestión del estado de la alarma y su interacción con el entorno.
- El diseño de un circuito limpio, eficiente y fácilmente replicable.

En resumen, el prototipo desarrollado responde satisfactoriamente a los requerimientos de una solución domótica básica, demostrando la viabilidad de construir sistemas inteligentes desde plataformas accesibles como Arduino. Este trabajo sienta las bases para futuras ampliaciones como integración con redes inalámbricas, monitoreo remoto, control desde dispositivos móviles o incorporación de más sensores.



## Bibliografía

- Arduino. (2024). *Arduino Language Reference*. Arduino Documentation. <https://docs.arduino.cc/>
- SparkFun Electronics. (2005). *TMP36: Low Voltage Temperature Sensor*. SparkFun. <https://www.sparkfun.com/datasheets/Sensors/TMP36.pdf>
- Parallax Inc. (2022). *PIR Sensor Rev B - Technical Documentation*. <https://www.parallax.com/downloads/pir-sensor-rev-b-documents/>
- Adafruit. (2023). *Using a LiquidCrystal I2C display*. Adafruit Learning System. <https://learn.adafruit.com/character-lcds>
- Autodesk Inc. (2024). *Tinkercad Circuits: Simulation and Prototyping Platform*. <https://www.tinkercad.com/>
- González, M., & Castillo, J. (2021). *Domótica aplicada con Arduino: Introducción práctica a sistemas inteligentes*. Alfaomega Grupo Editor.
- Universidad Don Bosco. (2024). *Guía de desarrollo de proyectos - Fase 2. Arquitectura de Computadoras*. Facultad de Ingeniería.