**INFORME PROYECTO**



**Sistema de confort térmico inteligente en aula de clases para dos estudiantes**

**Elaborado por:**

**EDWARD ESTEBAN DÁVILA SALAZAR**

**104623010331**

**JUAN JOSE HURTADO MOLANO**

**104622020695**

**Universidad del Cauca**

**Facultad de Ingeniería Electrónica y Telecomunicaciones**

**Ingeniería de Sistemas**

**Arquitectura Computacional**

**Popayán, Cauca, Colombia**

**14-10-2025**

Contenido

[Sistema de confort térmico inteligente en aula de clases para dos estudiantes 1](#_Toc211457470)

[Planteamiento del problema 1](#_Toc211457471)

[Diagrama General del proyecto 1](#_Toc211457472)

[Descripción de cada módulo que conforma el proyecto 2](#_Toc211457473)

[Diagrama de flujo del software 3](#_Toc211457474)

[Herramientas de desarrollo 4](#_Toc211457475)

[Componentes software 5](#_Toc211457476)

[Interfaz gráfica de usuario (GUI) 5](#_Toc211457477)

[Sensores 5](#_Toc211457478)

[Actuadores 5](#_Toc211457479)

[Alarmas 5](#_Toc211457480)

[Proceso de pruebas 5](#_Toc211457481)

[Test Backlog (min 10 pruebas) 6](#_Toc211457482)

[Diagrama final 6](#_Toc211457483)

[Fotografías evidencias (min 5) 7](#_Toc211457484)

[Problemas encontrados (min 4) 8](#_Toc211457485)

[Link GitHub 9](#_Toc211457486)

# Sistema de confort térmico inteligente en aula de clases para dos estudiantes

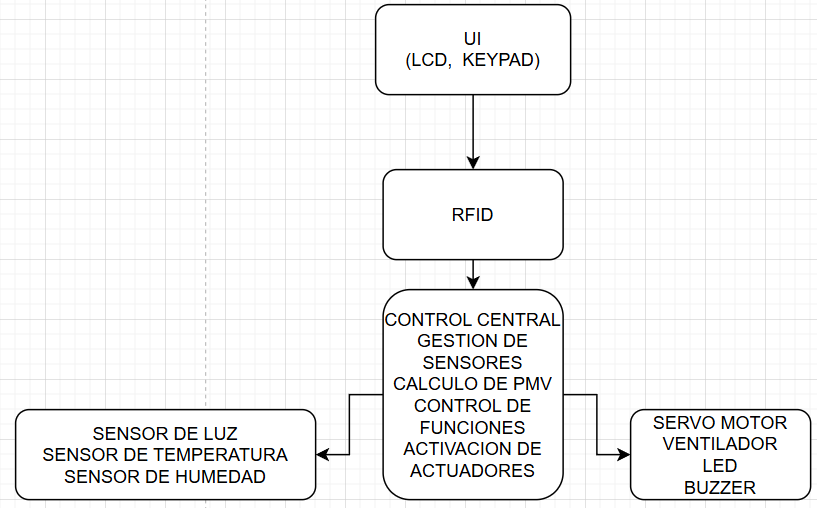
# Planteamiento del problema

En entornos educativos cerrados, el confort térmico de los ocupantes influye sobre su comodidad, capacidad de atención y rendimiento cognitivo. En aulas pequeñas (2 estudiantes), variaciones en temperatura, humedad relativa e iluminación pueden provocar condiciones de incomodidad térmica que requieren acción correctiva. La monitorización manual o la intervención humana no siempre son factibles; por ello se propone un sistema embebido capaz de:

* Medir variables ambientales relevantes (temperatura, humedad relativa y luz).
* Evaluar una métrica de confort térmico (PMV, Predicted Mean Vote) mediante una tabla de consulta.
* Ejecutar actuadores (ventilador, servomotor) o alarmas cuando el confort esté fuera de los umbrales definidos.
* Implementar control de acceso y configuraciones mediante seguridad por clave y registro de TAG RFID.

El objetivo es diseñar, implementar y documentar un prototipo que automatice el mantenimiento de condiciones de confort para los usuarios del aula.

# Diagrama General del proyecto



# Descripción de cada módulo que conforma el proyecto

Seguridad (Inicio)

* Función: solicitar clave de 4 dígitos vía keypad.
* Comportamiento: clave correcta → Config; clave incorrecta → Bloqueado (hasta pulsar botón físico).
* Componentes: Keypad, LCD, botón, LED rojo.

Config (Registro TAG)

* Función: leer TAG RFID autorizado y guardar bandera de autorización en EEPROM.
* Comportamiento: TAG válida → esperar 5 s → Monitor.
* Componentes: RC522, EEPROM, LCD.

Monitor (Lectura y evaluación)

* Función: leer sensores (NTC, DHT11, LDR), calcular PMV por interpolación en tabla (LUT), mostrar datos en LCD.
* Comportamiento: cada 2.5 s se realiza un ciclo de lectura/decisión; aplica ventana de reposo para evitar oscilaciones.
* Componentes: NTC (A0), DHT11, LDR (A1), LCD.

PMV Alto

* Condición: PMV > 0.5.
* Acción: activar relay → ventilador (12 V), parpadeo LED rojo (300 ms ON / 100 ms OFF) durante 3 s.

PMV Bajo

* Condición: PMV < −0.7.
* Acción: girar servo 90° (simulación de apertura/acción), parpadeo LED azul (100 ms ON / 400 ms OFF) durante 2 s.

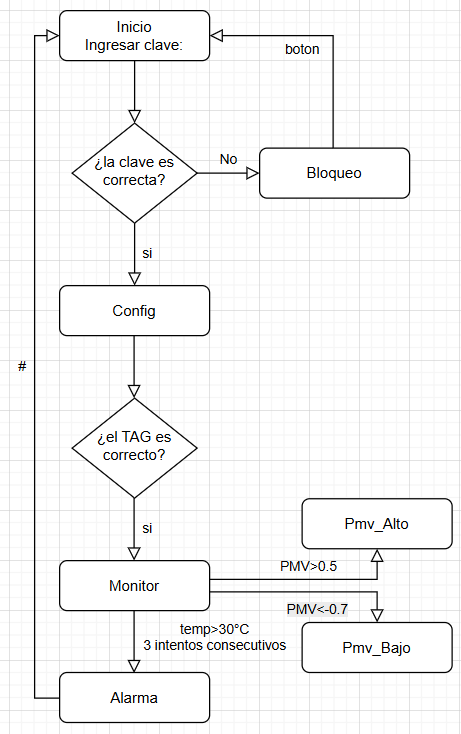
Alarma

* Condición: temperatura > 30 °C durante 3 lecturas consecutivas.
* Acción: buzzer y parpadeo LED rojo (100/500 ms). Se desactiva al presionar tecla # y regresa al estado Inicio.

Bloqueado

* Condición: ingreso de clave incorrecta.
* Acción: muestra BLOQUEO y requiere pulsación de botón físico para regresar a Inicio.

# Diagrama de flujo del software



# Herramientas de desarrollo

Para el desarrollo del proyecto las herramientas que se están usando son:

|  |  |
| --- | --- |
| Wokwi. Simulación de circuitos y programación en Arduino. | Wokwi - YouTube |
| Repositorio en GitHub. Manejo de versiones de código fuente. |  |
| **Doxygen:** herramienta de documentación automática del software. | Doxygen: Comprehensive Software Documentation Guide |
| **Arduino IDE:** desarrollo y carga del código fuente | Arduino en la programación y robótica educativa – Observatorio del Gabinete  de Tele-Educación |

# Componentes software

## Interfaz gráfica de usuario (GUI)

| Librería | Función | Descripción | Estado |
| --- | --- | --- | --- |
| Keypad | char getKey() | Devuelve la tecla presionada. | Implementado y probado |
| LiquidCrystal | lcd.print() | Muestra mensajes en pantalla. | Implementado y probado |

## Sensores

| Librería | Función | Descripción | Estado |
| --- | --- | --- | --- |
| DHT | readHumidity(), readTemperature() | Lee temperatura y humedad del ambiente. | Implementado y probado |
| analogRead() | Lectura de NTC y LDR | Captura temperatura por NTC y luz por LDR. | Implementado y probado |

## Actuadores

| Librería | Función | Descripción | Estado |
| --- | --- | --- | --- |
| Servo | servo.write() | Controla la posición del servomotor. | Implementado y probado |
| digitalWrite() | Control del relay y LEDs. | Activa/desactiva ventilador o indicadores. | Implementado y probado |

## Alarmas

* Alarma por temperatura alta: activada cuando la temperatura supera 30 °C en 3 lecturas consecutivas; emite buzzer y LED rojo intermitente.
* Bloqueo por intentos fallidos: después de ingreso de clave incorrecta, sistema bloqueado hasta pulsar botón físico.

# Proceso de pruebas

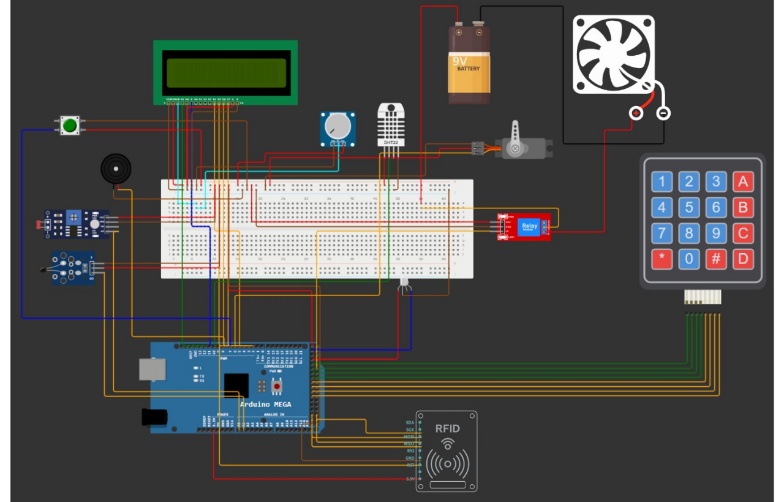
Se siguió una metodología basada en cuatro cuadrantes:

1. Pruebas unitarias: validación de lectura de sensores y respuesta de actuadores.
2. Pruebas de componentes: integración de sensores con la interfaz LCD.
3. Pruebas de integración: evaluación completa del sistema FSM en funcionamiento.
4. Pruebas de desempeño: observación de la estabilidad del sistema bajo funcionamiento continuo.

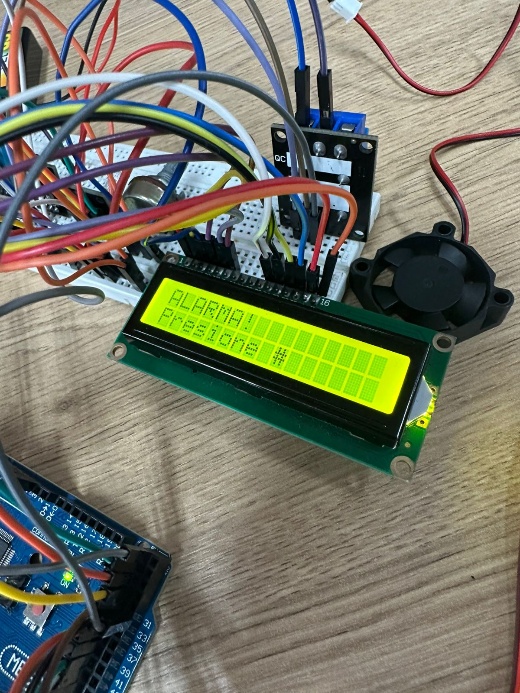
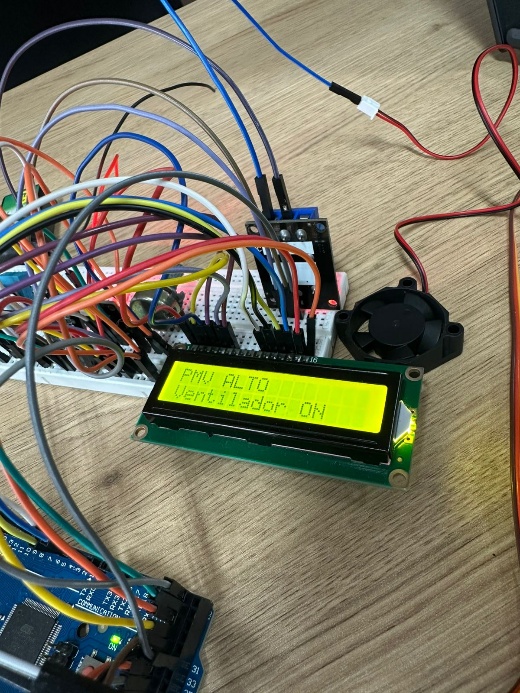
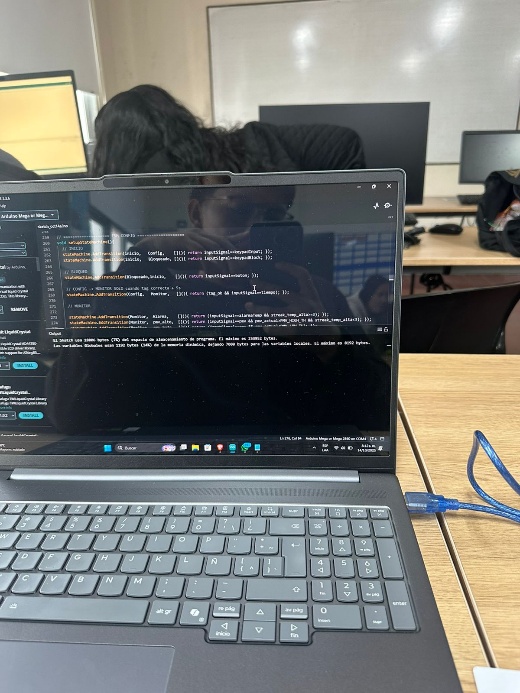
## Test Backlog (min 10 pruebas)

| # | Prueba | Descripción | Pasó | Observaciones |
| --- | --- | --- | --- | --- |
| 1 | Lectura de temperatura NTC | Verificar rango 20°C–30°C | Sí | Estable |
| 2 | Lectura de humedad DHT11 | Variar con humedad de la sala vs ambiente exterior | Sí | Precisión ±3% |
| 3 | Lectura de luz LDR | Cubrir y destapar sensor | Sí | Responde bien |
| 4 | Ingreso de clave | Introducir clave incorrecta | Sí | Sistema bloquea |
| 5 | RFID válido | Acercar TAG registrada | Sí | Transición correcta |
| 6 | PMV alto | Simular T>32°C, RH<50% | Sí | Ventilador activo |
| 7 | PMV bajo | Simular T<20°C, RH>80% | Sí | Servo activado |
| 8 | Alarma | Forzar T>30°C por 3 lecturas | Sí | Buzzer y LED activos |
| 9 | Reinicio de bloqueo | Pulsar botón físico | Sí | Regresa a Inicio |
| 10 | Visualización LCD | Confirmar actualización cada ciclo | Sí | Información clara |

# Diagrama final



# Fotografías evidencias (min 5)



# Problemas encontrados (min 4)

* Componentes físicos en mal estado → se hizo el cambio para asegurar la conexión.
* Variación en lectura del NTC por mal contacto en protoboard → se mejoró conexión.
* DHT11 arrojaba NaN en lecturas rápidas → se agregó retardo de 2.5s por ciclo.
* Interferencia entre RFID y relay → se usó filtrado y conexión a GND común.

# Link GitHub

<https://github.com/DVLASZ/Proyecto-Comfort-Termico>