

Especificación del Sistema de Monitoreo de Confort Térmico

Versión: 1.0

Fecha: 24 de Diciembre de 2025

Estado: Implementado

1. Introducción

1.1. Propósito del Sistema

El propósito del sistema es permitir el monitoreo participativo del confort térmico en espacios educativos (aulas). A través de una aplicación web accesible vía código QR, los ocupantes pueden reportar su sensación térmica en tiempo real. Estos datos se agregan para permitir a los administradores visualizar el estado de confort de las instalaciones y tomar decisiones informadas.

1.2. Alcance

El sistema abarca:

- **Interfaz de Usuario (Frontend):** Aplicación web progresiva para votación y gestión de perfil de usuario.
- **Backend as a Service (BaaS):** Gestión de base de datos, autenticación y tiempo real

2. Arquitectura del Sistema

2.1. Stack Tecnológico

- **Frontend:**
 - **Lenguajes:** HTML5, CSS3, JavaScript (ES6+).
 - **Framework CSS:** Tailwind CSS (vía CDN).
 - **Librerías:** Chart.js (Gráficas), Supabase JS Client.
- **Backend:**
 - **Base de Datos:** PostgreSQL (alojado en Supabase).
 - **Autenticación:** Supabase Auth.
 - **Tiempo Real:** Supabase Realtime (WebSockets).
- **Infraestructura:**
 - **Hosting:** Netlify (Archivos estáticos).

3. Actores y Roles

3.1. Usuario Anónimo

- Puede acceder a la aplicación.
- Puede seleccionar un aula.
- Puede emitir votos de confort.
- **Limitación:** No acumula puntos y no tiene historial persistente entre dispositivos.

3.2. Usuario Registrado

- Todas las capacidades del usuario anónimo.
- Tiene credenciales de acceso (Email/Contraseña).
- Acumula puntos por cada voto (+10 pts).
- Aparece en el Ranking de usuarios.
- Puede recuperar su contraseña vía email

4. Especificación de Requisitos Funcionales

4.1. Módulo de Autenticación y Usuarios

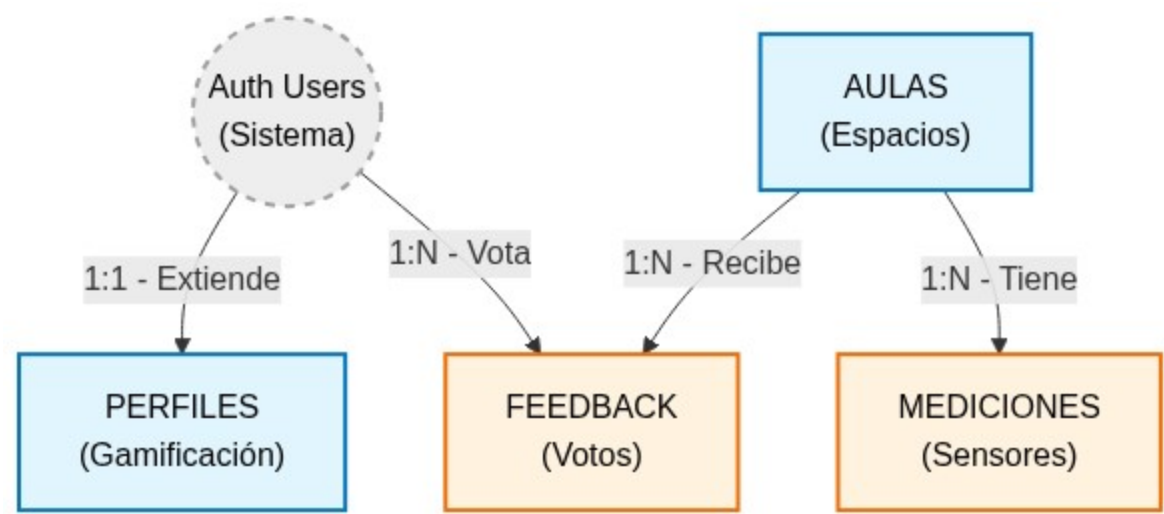
- **Registro/Login:** Se utiliza autenticación por Email y Contraseña. Se ha desactivado la confirmación de email para agilizar el registro.
- **Recuperación de Contraseña:** Implementado mediante flujo SMTP usando Gmail. El usuario recibe un enlace, hace clic y define una nueva clave.
- **Gestión de Perfil:** Los usuarios logueados pueden cambiar su contraseña desde la interfaz principal.

4.2. Gestión de Espacios (Aulas)

- **Selección de Aula:**
 - **Automática:** Vía parámetro URL `?aula=ID` (QR).
 - **Manual:** El usuario introduce el ID numérico.

5. Modelo de Datos (Base de Datos)

5.1. Diagrama Entidad-Relación



5.2. Tablas Principales

`public.aulas`

Catálogo de espacios disponibles.

Campo	Tipo	Descripción
-------	------	-------------

6. Diseño de Interfaz (UI/UX)

6.1. Pantalla Principal (`index.html`)

- **Cabecera:** Estado del usuario (Puntos, Email, Botón Salir/Cambiar Clave) o Botón de Login.
- **Estado Aula:** Muestra el aula actual o alerta "Sin Aula Asignada".
- **Botonera de Voto:** 5 botones grandes con emojis representativos y colores semánticos (Azul a Rojo).
- **Modales:** Ventanas emergentes para Login/Registro, Selección de Aula, Recuperación de Contraseña y Cambio de Contraseña.

6.2. Pantalla Admin (`admin.html`)

- **Bloqueo:** Pantalla inicial solicitando contraseña.
- **Dashboard:** Diseño en rejilla (Grid) responsive.

7. Configuración y Despliegue

7.1. Configuración SMTP (Gmail)

Para evitar límites de envío, se utiliza un servidor SMTP personalizado:

- **Host:** `smtp.gmail.com`
- **Port:** `587` (SSL OFF / STARTTLS).
- **Auth:** Usuario de Gmail y Contraseña de Aplicación (App Password).

7.2. Scripts de Mantenimiento

El sistema cuenta con scripts SQL predefinidos para gestión:

- `reset_full_database.sql` : Reinicio completo de fábrica.
- `add_aulas.sql` : Generación masiva de aulas (1-50).
- `allow_public_profiles.sql` : Configuración de permisos para el ranking.

8. Accesos y Credenciales

8.1. Direcciones Públicas

- Aplicación de Voto (Usuarios): <https://control-termico.netlify.app>
- Panel de Administración (Dashboard): <https://control-termico.netlify.app/admin.html>

8.2. Credenciales de Administración

- Contraseña de acceso al Dashboard: `admin1234`
 - *Nota: Esta contraseña se valida en el cliente (`admin.js`) y sirve para restringir la visualización de datos sensibles.*