



Ambiental App

Informe Técnico

Ambiental App es un sistema desarrollado con el objetivo de monitorear en tiempo real las condiciones ambientales en salones de clase, mejorando el confort térmico y lumínico de los espacios educativos. El sistema se basa en la recolección de datos mediante sensores físicos conectados a un ESP32 y visualización en una aplicación móvil y una plataforma web, alineado con las normas internacionales ASHRAE 55, ISO 7730 e ISO/CIE 8995-1.

Erik Jesith López Erazo

Victor Hugo Tejada Ricardo

Desarrollo de software

Contenido

Contenido	2
Informe técnico software Ambiental App	3
Desarrollo plataforma web de Ambiental App.....	4
Desarrollo aplicación móvil Ambiental App	5
Arquitectura del Sistema	7
Tecnologías usadas	8
Bibliografía.....	8

Informe técnico software Ambiental App

El desarrollo del software Ambiental App tiene como propósito brindar una solución tecnológica para el monitoreo y gestión en tiempo real de condiciones ambientales como la temperatura, humedad y luminosidad en salones de clase, con el fin de contribuir al confort y bienestar de los estudiantes y docentes, así como al cumplimiento de normas internacionales de calidad ambiental.

Este sistema se diseñó como una herramienta de apoyo para las instituciones educativas, permitiendo tomar decisiones oportunas frente a condiciones ambientales no óptimas.

El software consta de dos componentes principales:

- Una aplicación móvil, que a partir del usuario que inicie sesión permite acceder a diferentes funcionalidades según su rol. Por ejemplo, algunos usuarios podrán visualizar únicamente los datos en tiempo real, mientras que otros tendrán acceso a funciones adicionales como edición de configuraciones, gestión de alertas o administración de usuarios.
- Una plataforma web, destinada exclusivamente para la visualización de datos en tiempo real e históricos desde un navegador, sin funciones administrativas ni de edición.

Desarrollo plataforma web de Ambiental App

La plataforma web de Ambiental App fue desarrollada como un proyecto independiente utilizando Flutter Web, lo que permitió crear una interfaz ligera, moderna y responsiva adaptada para navegadores. Esta plataforma se diseñó específicamente con fines de visualización, brindando a los usuarios acceso a datos ambientales en tiempo real y al historial de lecturas generadas por los sensores.

Aunque el código fuente de la versión web es distinto al de la aplicación móvil, ambas plataformas están conectadas al mismo proyecto en Firebase, compartiendo recursos como:

- Firebase Realtime Database (para lectura de datos en vivo).
- Firebase Firestore (para historial y alertas).
- Firebase Authentication (para el control de acceso).
- Firebase Hosting (para la publicación de la web).

La arquitectura de la plataforma web se basa en el uso directo del SDK de Firebase para Flutter, lo que permite una integración eficiente y sin intermediarios con los servicios en la nube. Gracias a esto, no fue necesario desarrollar un backend propio o una API adicional para la sincronización de datos.

El despliegue de la plataforma web se realizó a través de Firebase Hosting, un servicio optimizado para el rendimiento y la seguridad, ideal para aplicaciones desarrolladas en Flutter Web.

El objetivo de la versión web es permitir a los usuarios consultar datos ambientales desde cualquier navegador con conexión a internet, sin necesidad de instalar una app. No se contemplan funciones de edición, administración ni cambios de configuración, ya que dichas funciones están disponibles exclusivamente desde la aplicación móvil según el rol asignado al usuario.

Desarrollo aplicación móvil Ambiental App

El desarrollo de la aplicación móvil de Ambiental App se realizó utilizando el framework Flutter, basado en el lenguaje de programación Dart, permitiendo construir una interfaz moderna, responsiva y con alto rendimiento para dispositivos Android. La aplicación fue diseñada para funcionar a partir de la versión Android 6.0 (API 23) en adelante.

A lo largo del proceso de desarrollo se siguieron buenas prácticas de arquitectura de software móvil, organizando el código mediante una estructura basada en el patrón MVVM (Modelo–Vista–ViewModel). Este enfoque facilita la separación entre la lógica de negocios, la interfaz de usuario y la gestión del estado de la aplicación, mejorando la mantenibilidad y escalabilidad del proyecto.

La aplicación permite funcionalidades diferenciadas según el rol del usuario que inicie sesión. Por ejemplo:

- Super Administrador y Administrador: pueden acceder a funciones de configuración de sensores, visualización de datos, historial, gestión de alertas y edición de roles.
- Usuario Final: tiene acceso limitado a visualización de datos y edición de su perfil personal.
- Encargado: puede revisar alertas y agregar comentarios en situaciones específicas.

La conexión con el backend se realiza directamente a través del SDK de Firebase para Flutter, permitiendo el uso de servicios como:

- Firebase Authentication: para el inicio de sesión y gestión de usuarios.
- Firebase Realtime Database: para la visualización de datos en tiempo real.
- Firebase Firestore: para el almacenamiento de historial de lecturas y alertas generadas.
- Firebase Storage (opcional): para el manejo de archivos o imágenes si es requerido.

Para pruebas y simulación durante el desarrollo, se utilizó Android Studio, configurando dispositivos virtuales con diferentes versiones del sistema operativo y tamaños de pantalla, garantizando la compatibilidad y experiencia de usuario fluida.

La app móvil representa el núcleo operativo del sistema, ya que es desde esta plataforma donde se gestionan configuraciones clave y donde se centraliza la experiencia del usuario final en el entorno educativo.

Arquitectura del Sistema

Backend y Base de Datos

- Base de datos: Firebase
- Servicios utilizados de Firebase:
- Realtime Database (datos en vivo desde sensores)
- Firestore (registro histórico y alertas)
- Firebase Authentication (control de acceso)
- Firebase Hosting (solo para la web)
- Firebase Storage (opcional para guardar recursos)

Aplicación Móvil

- Lenguaje: Dart (Flutter)
- IDE: Visual Studio Code
- Simulación: Android Studio
- Compatibilidad mínima: Android 6.0 (API 23 - Marshmallow)
- Arquitectura: MVVM simplificado
- Funcionalidades:
- Visualización de datos en tiempo real
- Alertas automáticas por fuera de los rangos
- Consulta de datos históricos
- Gestión de usuarios con distintos roles

Plataforma Web

Framework: Flutter Web

Hospedaje: Firebase Hosting

Función principal:

- Visualización únicamente de datos ambientales en tiempo real e históricos, sin capacidad de edición ni administración.
- Orientada a usuarios que solo requieren monitoreo desde escritorio.

Herramienta / Servicio	Función
Visual Studio Code	Entorno de desarrollo para móvil y web
Android Studio	Emulador y pruebas de la app móvil
Flutter	Framework usado en ambas plataformas
Firebase Realtime Database	Visualización en tiempo real
Firebase Firestore	Registro y análisis histórico de lecturas
Firebase Authentication	Control de acceso y gestión de usuarios
Firebase Hosting	Hospedaje de la plataforma web
ESP32	Dispositivo de adquisición de datos con sensores

Bibliografía

- ASHRAE. (2017). *ASHRAE. Standar 55*. Retrieved from Thermal Environmental Conditions for Human Occupancy. American society of Heating, Refrigerating and Air-conditioning Engineers: <https://www.ashrae.org>
- Developers, G. (2023). *Firebase for flutter*. Retrieved from FlutterFire: <https://firebase.flutter.dev/>
- GlobalSign, G. (2024). *IoT for all*. Retrieved from IoT and Environmental Monitoring with Sensor Networks: <https://www.iotforall.com/iot-and-environmental-monitoring-with-sensor-networks>
- Illumination, I. C. (2023). *ISO/CIE 8995-1*. Retrieved from Lighting of Work Places - Part 1: Indoor: <https://cie.co.at/publications/light-and-lighting-lighting-work-places-part-1-indoor>
- Standardization, I. O. (2005). *ISO 7730: Ergonomics of the thermal environment*. Retrieved from Analytical determination and interpretation of thermal comfort using calculation of the PMV and PPD indices and local thermal comfort criteria: <https://www.iso.org/standard/39155.html>