

# Proyecto IoT

Sebastián Vargas-Díaz, María Paula Romero-Garro, Yovelky María Delgado-Solís

26 de octubre de 2025

## 1. Introducción

El presente proyecto tiene como objetivo el desarrollo de una solución IoT orientada a la consulta y análisis de variables ambientales. Dicha solución se enfoca en la captura, almacenamiento y visualización de datos meteorológicos mediante herramientas de software y servicios en la nube.

Para la generación de los datos ambientales se utilizó la API de OpenWeather, la cual proporciona información sobre variables como temperatura, humedad y radiación solar (índice UV). El almacenamiento de los datos se gestionó a través de Firebase, utilizado como base de datos en la nube para asegurar la persistencia, sincronización y disponibilidad de la información. De esta manera, los datos generados y procesados se pueden consultar de forma remota, posibilitando el desarrollo de una aplicación web interactiva que presenta los resultados de manera intuitiva mediante gráficos y estadísticas.

El proyecto busca no solo simular el comportamiento de un sistema IoT funcional, sino también ofrecer una herramienta práctica para la visualización y análisis de condiciones ambientales. A través de la integración de servicios web, procesamiento de datos en Python y almacenamiento en Firebase, se logró una solución completa, escalable y alineada con los principios de interconectividad, automatización y accesibilidad que caracterizan a la Internet de las Cosas

## 2. Generación y gestión de variables del entorno

Para la generación de las variables ambientales utilizadas en el presente proyecto (humedad, temperatura y radiación ultravioleta (UV)) se empleó la API de OpenWeather, una herramienta en línea que permite obtener datos meteorológicos actualizados de prácticamente cualquier ciudad del mundo, de manera gratuita y con alta fiabilidad.

Para utilizar esta API, fue necesario crear una cuenta en la plataforma y generar una clave personal (API Key), la cual habilita el acceso a los distintos servicios disponibles. Los valores de temperatura y humedad se obtuvieron directamente a partir de los datos proporcionados por la API, sin necesidad de procesamiento adicional.

Sin embargo, el caso de la radiación UV requirió un tratamiento distinto, ya que los valores entregados por OpenWeather no son consistentes con la escala estándar de radiación solar utilizada actualmente en meteorología, generando resultados con magnitudes superiores a las esperadas. Para corregir esta discrepancia, se aplicó un ajuste basado en la normalización de datos, consistente en multiplicar el valor original por 10 y posteriormente dividirlo entre 25, lo cual aproxima el resultado a una escala realista y coherente con los índices utilizados por organismos climatológicos.

Por otra parte, el pronóstico semanal de las variables fue obtenido mediante las funciones de previsión de la misma API, con excepción del índice de radiación UV, que se calculó mediante una función adicional desarrollada en Python. Dicha función utiliza parámetros como la nubosidad y otros factores atmosféricos para estimar el valor del índice, permitiendo así ofrecer una predicción más precisa y útil dentro del sistema IoT implementado.

Finalmente, para la generación del pronóstico mensual, no fue posible obtener los datos directamente desde la API, ya que dicha funcionalidad no se encuentra disponible en la versión gratuita del servicio. Por esta razón, los datos correspondientes al mes de octubre se generaron a partir de la captura continua de información real durante el desarrollo del proyecto, lo que permitió construir un conjunto de datos representativo y verosímil. Si bien estos valores no son exactos, reflejan condiciones ambientales auténticas del periodo, garantizando coherencia y utilidad en las visualizaciones del sistema.

### 3. Gestión y almacenamiento de datos en Firebase

Para el manejo y almacenamiento de la información generada por el sistema, se utilizó el servicio Firebase, una plataforma en la nube desarrollada por Google, que ofrece soluciones escalables y en tiempo real para el desarrollo de aplicaciones web y móviles. Su uso resulta conveniente debido a su facilidad de integración con múltiples lenguajes de programación, su bajo costo en las versiones básicas y su compatibilidad con proyectos basados en IoT.

El acceso a Firebase únicamente requiere disponer de una cuenta de Google, crear un proyecto dentro de la consola de Firebase, habilitar los servicios necesarios (en este caso, la base de datos en tiempo real) y obtener las credenciales correspondientes para establecer la conexión desde el entorno de desarrollo.

Mediante el código implementado en Python, se realizó la conexión con Firebase utilizando dichas credenciales, lo que permitió la lectura y escritura de datos de manera segura y estructurada. Esta comunicación se gestionó a través de las librerías oficiales de Firebase para Python, garantizando la autenticación y la sincronización adecuada entre el sistema local y la nube.

En cuanto al modelo de almacenamiento, se definió una estructura jerárquica basada en nodos únicos, donde cada nodo representa una sesión diaria de registro. Dentro de cada nodo se almacenan las variables ambientales capturadas ese día (temperatura, humedad y radiación UV), junto con su marca de fecha y hora correspondiente. Además, dentro del mismo nodo se guarda el pronóstico semanal generado en esa fecha, de modo que cada conjunto de datos conserva su contexto temporal completo.

Este enfoque permite una organización lógica y cronológica de la información, facilita la consulta de datos históricos y optimiza los procesos de visualización y análisis posteriores. Asimismo, la estructura modular empleada en la base de datos posibilita la escalabilidad del sistema, permitiendo integrar nuevas variables o extender el periodo de almacenamiento sin alterar el funcionamiento general del proyecto.

### 4. Generación de reportes en formato PDF

Para complementar la visualización y análisis de los datos generados, se implementó una funcionalidad que permite la creación de reportes en formato PDF, en los cuales se incluyen comparaciones entre las diferentes variables ambientales del proyecto, como la temperatura, la humedad y la radiación ultravioleta. Para lograrlo, se emplearon diversas bibliotecas de JavaScript que facilitan tanto la generación de gráficos dinámicos como la exportación de la información a un documento descargable.

Mediante el uso de estas herramientas, fue posible representar de manera visual las variaciones de las variables en distintos periodos de tiempo, permitiendo realizar comparaciones entre fechas o rangos específicos. Esta capacidad resulta especialmente útil para detectar tendencias, comportamientos anómalos o patrones ambientales, como cambios de temperatura a lo largo del día o variaciones en la radiación según las condiciones atmosféricas.

Una vez generadas las gráficas y tablas comparativas, el sistema permite guardar el análisis en un archivo PDF, lo que facilita la conservación de los resultados, su distribución y consulta posterior. Esta función de exportación contribuye significativamente al valor práctico del proyecto, ya que permite documentar los resultados obtenidos, crear registros históricos y brindar al usuario una herramienta clara y accesible para la interpretación de los datos. En conjunto, esta funcionalidad fortalece la integración entre la visualización interactiva y el almacenamiento de la información, ofreciendo una experiencia más completa y útil dentro del sistema IoT desarrollado.

### 5. Conclusiones

El desarrollo de este proyecto permitió comprobar la viabilidad de una solución IoT accesible y funcional, orientada al monitoreo y análisis de variables ambientales mediante una plataforma web responsive. A través de la integración de tecnologías modernas como Python, Firebase y la API de OpenWeather, se logró implementar un sistema capaz de obtener, procesar, almacenar y visualizar información meteorológica de manera automatizada, precisa y adaptable a distintos dispositivos.

La API de OpenWeather fue esencial para la generación de datos reales, proporcionando valores actualizados de temperatura, humedad y radiación UV. Gracias al procesamiento realizado en Python, fue posible corregir inconsistencias y aplicar cálculos adicionales para obtener valores ajustados a

las escalas utilizadas en meteorología. Además, se implementó un mecanismo propio para estimar la radiación UV en los pronósticos semanales, lo que permitió complementar la información que no estaba disponible en la versión gratuita de la API.

Por su parte, el uso de Firebase como base de datos en la nube garantizó un almacenamiento estructurado, seguro y eficiente de los datos. La conexión establecida mediante credenciales permitió registrar lecturas diarias y pronósticos semanales bajo una estructura jerárquica de nodos, asegurando coherencia temporal y facilitando la consulta histórica de la información. Este modelo de almacenamiento también permitió mantener la sincronización entre los datos del servidor y la aplicación web, mejorando la experiencia del usuario final.

Asimismo, la integración de funciones para generar reportes en formato PDF añadió un valor significativo al sistema. Esta característica permitió comparar las variables ambientales entre diferentes fechas y conservar la información mediante documentos descargables. De este modo, el usuario puede analizar tendencias, realizar seguimiento de los cambios climáticos y mantener un registro histórico de los datos de manera práctica y visual.

En conjunto, el proyecto logró un equilibrio entre funcionalidad técnica, diseño visual y accesibilidad multiplataforma. La implementación en una página web responsive asegura que el sistema pueda ser utilizado desde cualquier dispositivo sin perder calidad en la visualización ni en la interacción. Esto convierte la solución desarrollada en una herramienta intuitiva, educativa y escalable, que puede servir como base para futuras ampliaciones en el ámbito del monitoreo ambiental y la gestión inteligente de datos en la nube.