

# Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Estado de México

## Bitácora de Proyecto: Sistema de Monitoreo Ambiental

TC3006C. Inteligencia artificial avanzada para la ciencia de datos

Grupo: 101

A01750908. Yóse Miguel Sotomayor Carneado

Prof. David Higuera Rosales

Fecha de entrega: 21 de septiembre de 2025

Semestre: agosto - diciembre 2025

### Objetivos

El propósito principal de esta práctica fue integrar hardware con sensores, almacenamiento en bases de datos, un backend con API REST y un modelo de Machine Learning conectado a una aplicación móvil. La meta fue construir un sistema capaz de recolectar datos ambientales, procesarlos y visualizarlos en tiempo real, generando además predicciones y alertas confiables.

### Materiales

Para la implementación del sistema se emplearon:

- ESP32-C3 DevKit (conectividad Wi-Fi nativa)
- Sensor DHT22 (temperatura y humedad)
- Sensor LDR con resistencia de 10 k $\Omega$
- LEDs indicadores con resistencias de 220 k $\Omega$
- Protoboard y cables jumper
- Resistencia pull-up para el DHT22

En cuanto a software:

- ESP32-C3 con ESP-IDF para firmware embebido
- FastAPI para el backend con integración de ML
- PostgreSQL para almacenamiento robusto de datos en series de tiempo
- Flutter para la aplicación móvil con gráficos interactivos y WebSocket en tiempo real

# Bitácora

## **Domingo, 8 de septiembre de 2025: Análisis inicial**

Definí junto con el equipo la arquitectura general del sistema, enfocándome en cómo garantizar la conexión fluida entre hardware, backend y app móvil.

## **Lunes, 9 de septiembre de 2025: Configuración de hardware**

Apoyé en la conexión de sensores al ESP32-C3 y en la validación de lecturas iniciales. Noté variaciones en condiciones reales, lo que motivó ajustes posteriores en firmware y en procesamiento de datos.

## **Martes, 10 de septiembre de 2025: Integración de firmware y flujo de datos**

Me enfoqué en diseñar un flujo de envío JSON eficiente hacia el backend, probando la estabilidad de la comunicación y reduciendo latencias en la transmisión de datos.

## **Miércoles, 11 de septiembre de 2025: API REST y almacenamiento**

Participé en la implementación de endpoints en FastAPI, con validación de datos y conexión a PostgreSQL. También identifiqué que la latencia en consultas históricas podía ser un cuello de botella, por lo que se optimizó con índices compuestos.

## **Jueves, 12 de septiembre de 2025: Modelos de Machine Learning**

Revisé los primeros entrenamientos del modelo (LightGBM), analizando el impacto de datos reales vs. simulados. El modelo mostró predicciones aceptables pero con ruido propio de la recolección en campo.

## **Viernes, 13 de septiembre de 2025: App móvil y visualización**

Probé la integración de gráficos en Flutter con datos en tiempo real. Detecté un lag inicial, que se mitigó con lazy loading y decimación de datos.

## **Sábado, 14 de septiembre de 2025: Alertas LED y notificaciones**

Validé el sistema de alertas tanto en hardware (LEDs por umbral de temperatura) como en software (notificaciones push en la app).

## **Domingo, 15 de septiembre de 2025: Pruebas de integración real**

Observé diferencias claras entre el comportamiento simulado y el real: las pruebas controladas eran estables, pero en ambiente real hubo ruido eléctrico y fluctuaciones de red Wi-Fi. Se justificaron con técnicas de preprocesamiento y reconexión automática.

## **Martes, 17 de septiembre de 2025: Documentación y reflexiones**

Redacté mis observaciones, justificaciones y conclusiones, enfocadas en la resiliencia del sistema y en la importancia de mantener la calidad de datos para modelos ML.

# Observaciones

Mi participación se centró en asegurar el flujo de datos entre hardware, backend y aplicación móvil. Identifiqué variaciones en sensores al trabajar en condiciones reales y observé que la latencia en la transmisión podía afectar el rendimiento general si no se optimizaba.

## **Justificaciones**

Las diferencias entre simulación y entorno real se explican por factores como ruido eléctrico, fluctuaciones de temperatura y problemas intermitentes de red. Mientras que en las simulaciones los datos eran estables, en campo fue necesario aplicar preprocesamiento, validación en la API y reconexiones automáticas para garantizar la calidad de la información.

## **Conclusiones**

Esta práctica me permitió reforzar la importancia de diseñar sistemas tolerantes a fallos y capaces de adaptarse a la realidad del entorno físico. El trabajo integró sensores, base de datos, modelos de Machine Learning y una aplicación móvil funcional, demostrando la relevancia de un flujo de datos bien estructurado. Concluyo que los sistemas IoT en ciencia de datos requieren no solo de componentes técnicos, sino también de estrategias de resiliencia y escalabilidad para ser realmente efectivos.