



# **Técnico superior en Telecomunicaciones**

## **Programación**

Proyecto: Visualizador de Temperatura y Humedad

Alumnos:

- Karina Jazmin Barbero.
- Macarena Aylén Carballo.
- Nicolás Nahuel Barrionuevo.
- Raúl Antonio Jara.

Profesor:

- Lisandro Lanfranco

ENTREGA N°2

## Índice

Resumen.....	2
Stack Tecnológico.....	2
Metodología .....	3
Conclusión.....	3

## Resumen del Proyecto

En la implementación de un dispositivo IoT para el monitoreo de variables ambientales, en este caso la temperatura y la humedad.

El sistema utiliza un sensor DHT11 para relevar ambas variables, las cuales son procesadas por un microcontrolador ESP32 que se comunica vía Wifi con una base de datos MySQL para almacenar los datos recogidos.

Se ha implementado un módulo LED WS2812 como actuador, que se activa cuando los valores de temperatura o humedad superan los umbrales predefinidos, proporcionando una señal visual de alerta.

Se ha creado una base de datos denominada **sensores**, que incluye dos tablas principales: **Mediciones** y **Actuador**. La tabla **Mediciones** almacena los datos obtenidos del sensor DHT11, incluyendo valores de temperatura y humedad junto con la fecha y hora de las lecturas. Por otro lado, la tabla **Actuador** registra el estado del LED WS2812, que se utiliza como actuador para visualizar alertas en función de los datos del sensor. Se han establecido umbrales para la temperatura y la humedad, y se implementaron funciones que insertan alertas en la tabla **Actuador** cuando las lecturas están fuera de rango. Este sistema permite un monitoreo efectivo de las condiciones ambientales y una respuesta visual inmediata a las variaciones en los datos.

## Stack Tecnológico Utilizado:

- **Sensor DHT11:** Utilizado para medir temperatura y humedad.
- **LED WS2812:** Actuador para la visualización de datos y alertas.
- **Base de Datos MySQL:** Para almacenar y gestionar los datos de mediciones y estados de actuadores.
- **MySQL Workbench:** Herramienta para la gestión de bases de datos y ejecución de scripts SQL.
- **Lenguaje de Programación (Python):** Para el desarrollo del código que interactuará con el sensor y la base de datos en tareas futuras.
- **Entorno de Desarrollo:** Visual Studio Code con PlatformIO: IDE y extensión para el desarrollo del código Python para la integración con la base de datos, depuración del firmware del ESP32, integrando librerías y configuraciones de hardware necesarias.

## Metodología de Trabajo

Para la creación de la base de datos **sensores**, seguimos un enfoque estructurado.

- Se definieron las tablas clave: **Mediciones**, para almacenar las lecturas de temperatura y humedad del sensor DHT11, y **Actuador**, para registrar el estado del LED WS2812 cuando las lecturas están fuera de los umbrales establecidos.
- Una vez definidas las tablas, se utilizó MySQL Workbench para ejecutar los scripts SQL correspondientes y verificar la estructura de la base de datos. Durante este proceso, se ajustaron permisos de usuarios para garantizar el acceso adecuado a la base de datos.

En cuanto al código en Python, la metodología fue basada en la modularización. Se implementó un script en Python que establece la conexión a la base de datos utilizando la biblioteca **mysql-connector-python**. A partir de esta conexión, se creó una función para insertar datos en la tabla **Mediciones**, simulando los valores del sensor DHT11. La funcionalidad de inserción fue probada y validada en MySQL para confirmar que los datos se almacenaban correctamente. Esta metodología permitió una integración clara y escalable entre el sistema de base de datos y el código Python, facilitando futuras ampliaciones del proyecto.

Este enfoque combina la planificación de la base de datos con una implementación eficiente en Python, garantizando una buena estructura para el manejo de los datos.

## Conclusión

En la segunda entrega del proyecto, se logró con éxito la creación de la base de datos **sensores**, con las tablas necesarias para almacenar los datos del sensor DHT11 y registrar el estado del actuador LED WS2812.

Además, se implementó un código en Python que permite la inserción de datos en la base de datos, asegurando una comunicación eficiente entre el sensor y el sistema de almacenamiento. Este avance proporciona una base sólida para futuras implementaciones, como la integración del microcontrolador y la gestión en tiempo real de las lecturas y alertas. Aún quedan por completar las siguientes fases del proyecto, que permitirán una recopilación de datos en vivo y la automatización del sistema, lo cual será abordado en las próximas entregas.

## Base de Datos sensores- Modelo Relacional

