

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Arquitectura de Computadores y
Ensambladores 2 B
Primer semestre 2025
Catedrático: Ing. Jurgén Andoni Ramírez Ramírez
Auxiliar: Axel Calderón
Auxiliar: Danny Cuxum



Proyecto Único - Fase 3

Sistema Inteligente de Monitoreo Ambiental para Cuartos de Servidores mediante IoT.

Objetivos:

- Diseñar un sistema de monitoreo inteligente para medir y registrar variables ambientales clave en un cuarto de servidores.
- Implementar un framework de IoT para la recolección y procesamiento de datos en tiempo real.
- Desarrollar un algoritmo de análisis de datos que interprete la información ambiental y emite alertas ante posibles anomalías.
- Crear una plataforma centralizada para la gestión y visualización de datos en dashboards intuitivos.
- Implementar un sistema de atención de datos mediante colas de mensajes (MQTT) para garantizar la integridad del análisis de datos.
- Desarrollar una interfaz para la visualización de datos, optimizada para el monitoreo y toma de decisiones en tiempo real.

Descripción General:

El presente proyecto propone un sistema de monitoreo ambiental para cuartos de servidores basado en tecnología IoT. Dado que los servidores son componentes críticos en infraestructuras de TI, es fundamental garantizar condiciones ambientales óptimas para su operación y evitar fallos debido a sobrecalentamiento, humedad excesiva o mala calidad del aire. El sistema está diseñado para recopilar datos sobre variables clave como temperatura, humedad, calidad del aire y presencia de personas en el entorno.

El sistema meteorológico y de control de acceso IoT recopila los datos de estos sensores y los envía a una plataforma centralizada implementando un sistema de cola de mensajes (MQTT). Los datos se almacenan en una base de datos y pueden visualizarse, implementando una aplicación web o herramientas que faciliten la construcción de dashboards. Esto permite que los usuarios monitoreen y accionen sobre los diversos componentes que administra el sistema, analizando las condiciones climáticas en tiempo real o realizando un seguimiento de tendencias a lo largo del tiempo.

Funciones:

Las funciones requeridas para determinar el funcionamiento correcto de las mediciones son:

- Medición de temperatura y humedad:
 - Los sensores miden y reportan la temperatura y humedad del cuarto de servidores, alertando sobre niveles críticos que puedan afectar el rendimiento de los equipos.
- Cantidad de luz en el ambiente:
 - El prototipo debe facilitar una entrada segura a los usuarios que visitan el cuarto de servidores, evitando la necesidad de prender un switch.
- Medición de calidad de aire:
 - Sensores de CO2 y partículas en el aire que permiten evaluar la calidad del entorno y prevenir problemas de ventilación o contaminación interna.

- Medición de proximidad
 - Sensores de movimiento y proximidad que identifican la presencia de personas en el cuarto de servidores, permitiendo registrar accesos y mejorar la seguridad.
- Sensor de corriente ACS712 5A
 - Sensores de corriente, deberá monitorear los niveles de energía que se están brindando a los servidores y data science, de forma que se registren si existen bajos niveles de energía o excesos.
- Motores y ventiladores
 - Deberá utilizar utilizar un motor servo motor simulando una entrada al cuarto de servidores
 - Ventiladores serán utilizados cuando las alertas de CO2 y Humedad son críticos.
 - Deberán simular algún ventilador con un motor DC esto para una alerta crítica de temperatura.
 - Al no detectar Energía Eléctrica deberán simular una alarma con un Buzzer/Zumbador.
- Sistema de alertas y notificaciones:
 - En caso de detectar niveles críticos de temperatura, humedad o calidad del aire, el sistema enviará alertas automáticas por la LCD
 - En casos de detectar niveles de Humedad, temperatura, Energía anormales deben alertarnos en la LCD y por medio de señales de advertencia mediante LEDS.
- Control RFID
 - Este módulo utiliza radiofrecuencia para leer una tarjeta de acceso, haciéndolo útil para sistemas de control de acceso donde Arduino puede tomar una acción.
- Análisis y proyección de Datos
 - El estudiante deberá utilizar la herramienta para el análisis de datos, lo cual se propuso en la fase 2
- Dashboard Web
 - El sistema mostrará la información mediante una aplicación

web sobre las mediciones históricas tomadas con los microcontroladores.

- El sistema mostrará información en tiempo real sobre las mediciones que se están tomando.
- El sistema mostrará proyecciones sobre factores climáticos.
- El sistema mostrará una mapa de calor de los datos.
- El sistema permitirá el acceso al cuarto de servidores por medio de un PIN al estar correcto se tomará una foto para la autenticación de doble factor.

- Base de datos

- El manejo de la información será constante, esto quiere decir que deben guardar datos, mostrar y manipularlos. Se recomienda utilizar alguna herramienta de Cloud para resguardar la información.

- Doble autenticación:

- Como medida de seguridad se deberá contar con la detección de rostro para que pueda acceder el personal autorizado al data center.

- Conectividad WIFI

- Realizar la conectividad entre los microcontroladores y el envío de los datos por medio de WIFI.

- MQTT

- Utilizar el protocolo MQTT para el envío de datos en tiempo real permitiendo que la información se mantenga de manera segura y rápida.

- Grafana

- Mantener un monitoreo constante a la base de datos mediante la herramienta de análisis de datos en tiempo real, en el cual deberán generar un análisis por cada tabla que posea en la base de datos, incluyendo la creación de un mapa de calor que permita visualizar de manera intuitiva las tendencias, patrones y valores críticos de los datos a lo largo del tiempo.

Fase 3

Introducción Fase 3

La fase 1 y fase 2 nos muestran cómo se pueden obtener los datos, transformarlos en información y cómo esta información puede trasladarse a una plataforma para ser visualizada por un usuario final.

Pero en la fase 3, concluimos con la integración de las cinco capas del IoT Stack Framework, presentando la posibilidad de visualizar, analizar, monitorear y predecir información, implementando datos históricos, albergados en una base de datos en la nube, además contando con la disposición de visualización amigable de data en tiempo real e histórica y generando alertas ante posibles cambios abruptos en la captura o manipulación de datos (implementando Grafana y la aplicación web).

Descripción de capas a implementar Fase 3

Hardware:

- Uso exclusivo del microcontrolador Arduino y placa Raspberry Pi.
- Se implementará lo realizado en la fase 1 y fase 2, para el análisis histórico y la recopilación de datos.
- Node MCU ESP8266 (conectividad Wi-Fi)

Software:

- Arduino IDE
- Raspberry PI SO
- Framework Web (React/Angular recomendados)
- Base de Datos (MySQL recomendado)
- API's (Python, Node.js, etc.)
- Reconocimiento facial (Face rekognition aws recomendado)

Plataforma:

- Pantalla LCD
- Dashboard en Aplicación Web
- VirtualBox (Pueden utilizar cualquier otro para crear VM)
- EMQX - MQTT
- Grafana

Despliegue en la Nube:

- Base de Datos
- Aplicación Web

- API con protocolo MQTT (**Ya no se permitirá mediante puerto serial a la PC**)

Descripción del funcionamiento

Presentación de Hardware:

Node MCU ESP8266

NodeMCU es una tarjeta de desarrollo similar a Arduino, especialmente orientada al Internet de las cosas (IoT). Está basada en el chip ESP8266EX, diseñado para cubrir las necesidades de un mundo conectado, mediante la conectividad Wi-Fi.

En nuestro caso, hemos venido utilizando el puerto serial desde el arduino a la computadora como medio de comunicación y enlace de datos, esto útil en un ambiente local y de pruebas, pero para finalizar nuestro proyecto simulando un escenario en producción, debemos de proveer la capacidad al arduino de conectarse directamente a Internet, para lo cual utilizaremos la conectividad que nos ofrece el NodeMCU ESP8266.

Dicho lo anterior, el envío de datos recolectados por el Arduino debe comunicarse con esta tarjeta, para manipular el traslado de datos por Internet mediante el protocolo MQTT (explicado más adelante) tanto para guardarlos en la base de datos, como para enviarlos a la aplicación web.

Nota: En caso de utilizar Raspberry Pi física, se puede emplear el módulo Wi-Fi integrado, tomando en cuenta que la comunicación debe ser directa desde Arduino a Raspberry, es decir, un cableado directo, sin uso de APIs externas que envíen datos hacia la Raspberry.

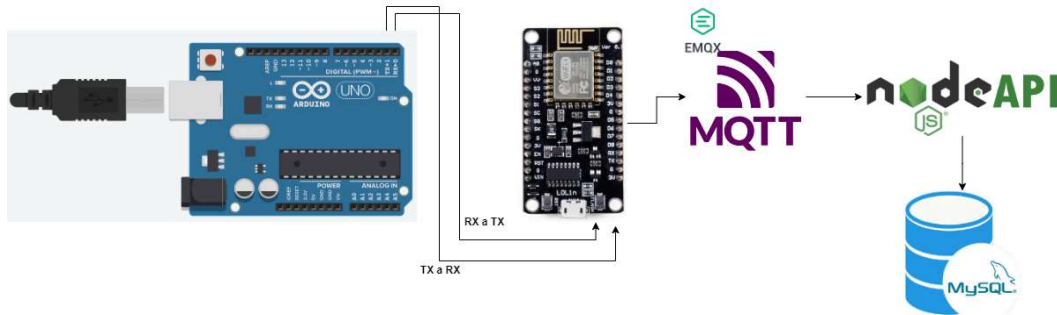
Presentación de información - Software:

Protocolo MQTT

El protocolo MQTT es un estándar de mensajería ligera que se utiliza comúnmente en aplicaciones de Internet de las cosas (IoT) debido a su bajo consumo de ancho de banda y energía. MQTT está basado en un modelo de "publicador/suscriptor", lo que significa que los dispositivos conectados pueden enviar (publicar) o recibir (suscribirse) mensajes a través de un servidor central conocido como "broker".

En nuestro proyecto, utilizaremos MQTT para facilitar la comunicación entre el NodeMCU ESP8266 y otros servicios o aplicaciones, como la base de datos y la aplicación web.

Almacenamiento de datos con MQTT:



Plataforma web con MQTT:

Permitirá visualizar los datos en tiempo real de todos los sensores, presentándose de manera atractiva y amigable (similar a la fase 1 con Processing).

Flujo de la información en la app:

Sensores y Motor -> Arduino -> MQTT -> plataforma web:

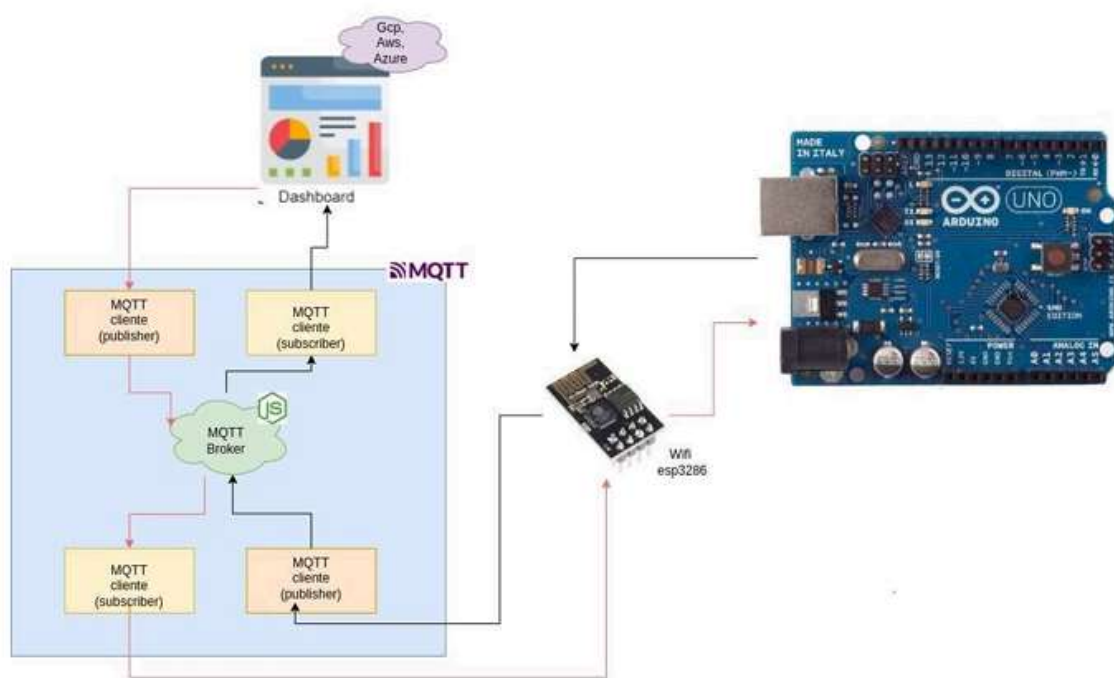
- Para visualizar los datos, el microcontrolador Arduino, debe enviar la información obtenida de los sensores, implementando conectividad Wi-Fi, para luego, enviar esta información a un cliente MQTT y con ello publicar un Topic específico en el Broker.
- Posteriormente la plataforma web, debe suscribirse a un Topic específico y mostrar la información resultante del Broker.
- Además, se requiere que cada vez que un usuario ingrese a los servidores, se envíe un mensaje/alerta a la aplicación, esto con el fin de que la aplicación quede prevista para ser utilizada también desde un enfoque administrativo o de seguridad del cuarto de servidores mediante la aplicación. Las alertas pueden ser simples *toasts* que se muestran durante un corto tiempo.

Plataforma web -> MQTT -> Arduino -> Motor, LCD y luces:

- La plataforma web, tendrá la capacidad de interactuar con la entrada al principal para lo que se emplea un factor de doble seguridad, la interacción se dará de la siguiente manera:
 - Se solicita que exista una opción de “Acceso al cuarto de Servidores” en la cual se ingrese un PIN, se valide y en caso

de ser correcto se le tomara una fotografía al usuario para validar que tenga la autorización de entrar al cumplir con el doble factor de seguridad se le envíe una señal al Arduino para que genere la acción de apertura mediante el Servomotor, indicando un mensaje en la pantalla LCD, de la misma manera, interactuar con el motor y la pantalla en caso el PIN/Rostro sea incorrecto (establecer pines por usuario en la base de datos, no es necesario crear un registro o login de usuarios).

- Establecer un botón para interactuar con la iluminación general del cuarto de servidores, que genere una acción de encendido y apagado en las luces.



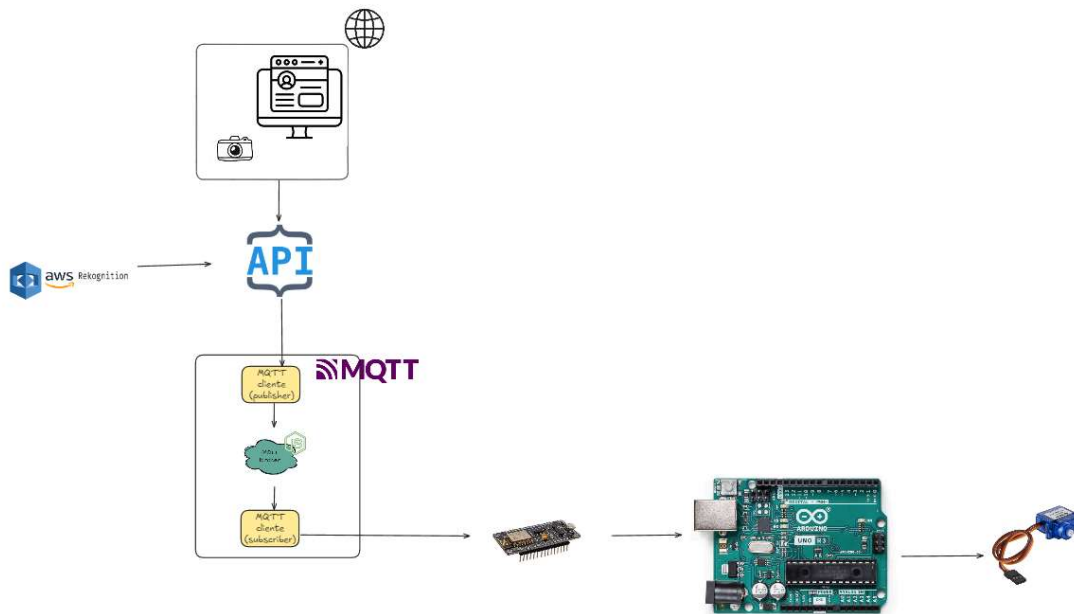
Explicación del reconocimiento facial

Face Rekognition

Amazon Rekognition es un servicio basado en inteligencia artificial diseñado para analizar imágenes y videos de manera rápida y precisa, destacando por su capacidad de identificar objetos, caras y texto. Este sistema utiliza algoritmos avanzados de aprendizaje automático para procesar el contenido visual y extraer información relevante. Rekognition ofrece funciones como el reconocimiento facial, que permite detectar y comparar rostros en imágenes, así como el análisis de emociones, identificación de celebridades.

En nuestro proyecto, face recognition se utilizará para hacer la autenticación de doble factor como medida de seguridad al entrar en el

data center.



El flujo de validación:

- Primero se validará que el PIN ingresado este correcto, luego en la API web se tomara una foto al personal que quiere entrar al data center
- Después la API web envía a la API del back para procesar la imagen.
- La foto se envía al servicio de reconocimiento facial(amazon rekognition/otro) para analizar si en la colección de imágenes está el rostro detectado.
- El servicio utilizado procesa la foto y devuelve los datos correspondientes a la página web(si detecta un rostro, o no si no está en la colección de rostros).
- La API web interpreta el resultado del reconocimiento facial, si la persona está en la colección y es detectada en un nivel de confianza(puesta por el estudiante) se genera una respuesta afirmativa. Si no lo está se genera una respuesta negativa.
- La API web publica el mensaje ("Sí" o "No") en el topic correspondiente del broker MQTT.
- El broker MQTT distribuye el mensaje a todos los suscriptores del topic.
- El Arduino, conectado mediante el módulo WiFi ESP8266, recibe el mensaje del broker MQTT.
- Basándose en la respuesta recibida, el Arduino acciona o no el servomotor, teniendo en cuenta que este proceso representa la segunda etapa de autenticación, posterior a una primera etapa.

Grafana

Grafana es una plataforma utilizada para la visualización y monitoreo de datos. Ofrece una interfaz amigable que permite crear paneles interactivos y personalizables, en los que se pueden visualizar métricas y datos provenientes de diversas fuentes, como bases de datos, sistemas de IoT o servidores.

En nuestro proyecto, Grafana se utilizará para monitorear los datos recolectados por el sistema IoT. Esto nos permitirá visualizar el flujo de datos en tiempo real, analizar el comportamiento del sistema y garantizar que la infraestructura IoT funcione correctamente, con un enfoque de usuario administrativo de la infraestructura del sistema y no tanto como usuario final.

Se requiere un panel con un indicador o gráfica para cada sensor (temperatura, humedad, calidad de aire, consumo de energía, etc.) y para el control de acceso al cuarto de servidores. Además, se implementará un mapa de calor en formato de calendario que permita visualizar de manera clara y organizada las tendencias y valores críticos de los datos a lo largo del tiempo (por días, semanas). Este mapa de calor será una herramienta clave para identificar patrones, anomalías o condiciones que requieran atención inmediata, optimizando así la toma de decisiones y la gestión del datacenter.

Ejemplo de gráfica:



Ejemplo de mapa calor:

Time Bucket	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
12:00 AM	0	0	0	0	0	0	0
1:00 AM	0	0	0	0	0	0	0
2:00 AM	0	0	0	1	0	0	0
3:00 AM	0	0	0	2	0	0	0
4:00 AM	0	0	0	0	0	0	0
5:00 AM	0	0	0	0	0	0	1
6:00 AM	1	1	1	0	0	0	1
7:00 AM	0	3	1	3	4	1	0
8:00 AM	0	6	7	23	8	2	1
9:00 AM	0	9	15	22	9	12	1
10:00 AM	0	7	13	13	10	9	0
11:00 AM	0	5	7	18	11	6	0
12:00 PM	0	7	18	16	8	7	0
1:00 PM	0	6	11	9	15	13	0
2:00 PM	0	12	12	8	12	8	0
3:00 PM	0	3	8	8	10	3	0
4:00 PM	0	3	5	5	7	3	0
5:00 PM	1	4	4	3	5	2	0
6:00 PM	0	2	0	1	1	1	0
7:00 PM	0	0	0	2	0	0	0
8:00 PM	0	1	0	0	3	1	0
9:00 PM	0	0	1	0	0	0	0
10:00 PM	0	1	0	1	0	0	0

Proyecciones Climáticas

En esta sección, los estudiantes pondrán en práctica los conceptos desarrollados en la fase 2, específicamente en lo que respecta a la proyección de datos climáticos. Su tarea consistirá en realizar proyecciones relacionadas con la propuesta presentada en la fase anterior.

Para entrenar el modelo de proyección, es fundamental contar con un conjunto mínimo de datos almacenados, de manera que las proyecciones tengan sentido y precisión. Se requerirá un mínimo de 10 días de datos climáticos para garantizar la calidad del modelo predictivo. Esta recolección de datos forma parte de la fase 3, sin embargo, dado que la fase 2 ya debería estar completada, la recolección de datos no debería verse afectada.

Una vez completada la recolección de datos, deberán asegurarse de que la propuesta tenga la capacidad de integrarse fácilmente con Python, ya que el modelo se ejecutará en una Raspberry Pi. Es crucial que el sistema funcione con una lógica adecuada, por lo que se requerirá que las proyecciones climáticas tengan un rango de entre 1 y 8 días. Por ejemplo, si el usuario está en lunes y desea conocer la proyección para el miércoles, el sistema debe proporcionar una respuesta precisa. Sin embargo, si la solicitud excede los 8 días de antelación, el sistema debe indicar que no es posible realizar la proyección.

La visualización de los resultados debe ser simple y clara para el usuario final. Por ello, los estudiantes deberán desarrollar un dashboard intuitivo que permite visualizar las proyecciones de manera atractiva y comprensible. El dashboard debe ofrecer opciones claras, como la posibilidad de seleccionar el día para el cual se desea realizar la proyección, y generar una visualización de los datos de manera creativa y visualmente interesante. A continuación se presenta un ejemplo de cómo podría visualizarse el funcionamiento de dicha proyección.



Como pueden verificar hay una opción de rango de fecha lo que permite que todas las graficas se acoplen y muestren los datos.

Diagrama Fase 2

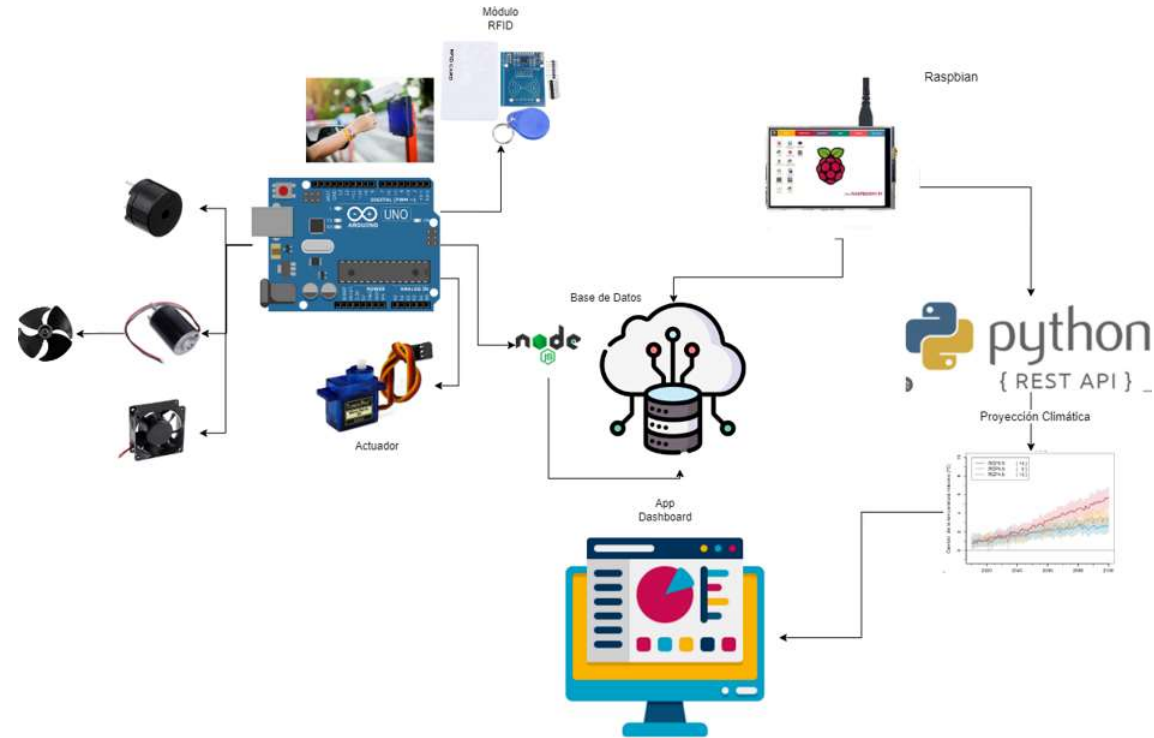
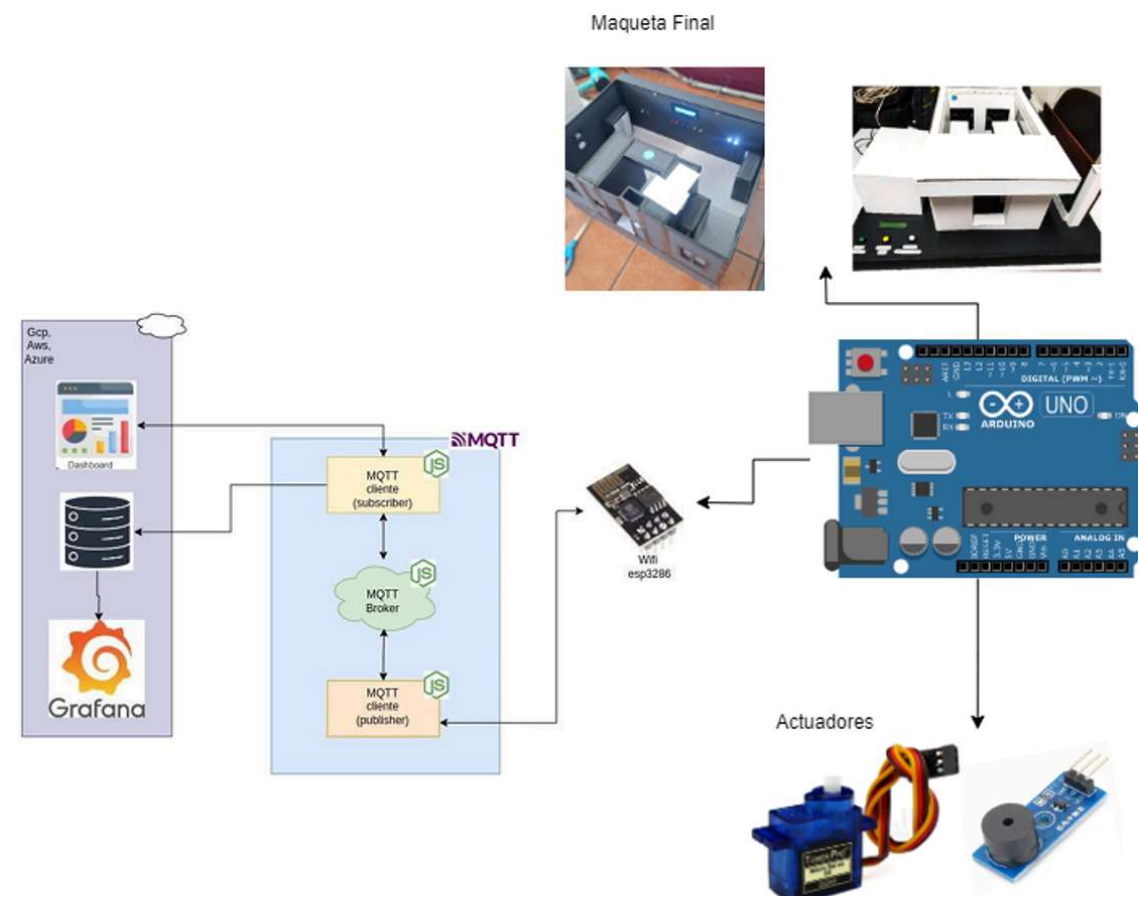


Diagrama Fase 3



Entrega:

Repositorio de GitHub.

Todo el código utilizado y la documentación deberá ser subido a un repositorio de github y al momento de la entrega sólo se mandará la documentación la cual deberá contener el link del repositorio, esto con el fin de evitar inconvenientes por el tamaño de los archivos al momento de la entrega, para la creación de dicho repositorio tomar en cuenta las siguientes indicaciones:

- Nombre del repositorio:
 - **ARQUI2B_2S2024_G<#GRUPO>**
 - Ejemplo: **ARQUI2B_1S2025_G5**
- Agregar el usuario del auxiliar como colaborador a su repositorio de github:
 - **AlexCB-16**
 - **DannyCSC**
- Todo código o documento que no se encuentre en el repositorio no será tomado en cuenta para la calificación.

Contenido obligatorio del repositorio:

- Código de Arduino.
- Código de APIs.
- Código de aplicación web.
- Documentación (explicación de código y funcionamiento de sensores, diagramas, componentes, en pdf o md).
- Es importante que la documentación describa las capas utilizadas del marco de trabajo Stack IoT Framework.

Estructura del repositorio:

Debido a que se usará el mismo repositorio durante todo el semestre, se solicita que contenga 3 carpetas, en las cuales se presente cada fase.

- Fase1
- Fase2
- Fase3

Nota: En el README del repositorio, colocar el número de grupo y los datos de sus integrantes.

Entregables:

- Realización de documentación: Se realizará la documentación correspondiente con detalles que destaquen el funcionamiento, usos, beneficios e impacto ambiental.
- Bocetos de prototipos: Bocetos de cada parte de realización de prototipos físicos con su explicación.
- **Maqueta final del prototipo propuesto:** Importante recordar que el enfoque general es sobre un **Sistema inteligente de Monitoreo Ambiental para Cuartos de Servidores mediante IoT**, por lo cual se necesita una maqueta física que encapsule los componentes utilizados en la fase y se oriente hacia el entorno del sistema.
- Descripción de las capas de Smart Connected design Framework utilizadas en la fase.
- Diagramas a criterios del estudiante que ayuden a evidenciar el flujo de la información.

Restricciones:

- Uso de Arduino, Raspberry PI SO (VM), Python, Base de Datos y Aplicación Web.
- Debe generar conexión VNC con arduino.
- Uso de conectividad Wi-Fi
- Uso de MQTT y Grafana
- La BD, App Web y APIs (excepto las que estén en Raspberry) deben estar desplegadas en la Nube.

Consideraciones:

- Todas las aclaraciones se realizarán en clase, por lo que deberán acatar todas las instrucciones escritas y verbales al momento de explicar el proyecto.
- Se calificará solamente lo que sea completamente funcional.
- Se deberán de enviar todos los entregables.
- **Las copias totales o parciales tendrán nota de cero y serán reportadas a la Escuela de Sistemas.**
- Fecha de entrega: **Fase 3: 02/05/2025**
- Fecha de calificación: **03/05/2025**

Aprobación de proyecto Fase 3

