Maestría en Ingeniería de Sistemas y Computación
Internet de Todas las Cosas

Autores:
Sebastián Murcia Gómez
Michael Dylan Blanquicett Carvajal
Oswaldo José Cáceres Leal
Juan David Martinez Moreno

DEFINICIÓN DEL CASO DE ESTUDIO

DEFINICIÓN DEL PROBLEMA

En el laboratorio de redes de la Universidad de los Andes, ubicado en el cuarto piso del edificio Mario Laserna, se ha identificado un uso subóptimo de 17 computadoras que permanecen encendidas durante todo el día, aunque el laboratorio solo es utilizado durante ciertos horarios de monitorias y clases de Infraestructura Computacional y cursos de maestría del departamento de Ingeniería de Sistemas y Computación. Además del consumo de energía innecesario, el laboratorio no cuenta con un sistema que regule la temperatura y humedad de los servidores, ni un control de acceso eficiente que garantice la seguridad y el uso adecuado del espacio. Esta situación genera un impacto negativo en términos económicos, medioambientales y de seguridad, afectando no solo el costo operativo, sino también la vida útil de los equipos y el desempeño de los servidores por las condiciones ambientales no controladas.

JUSTIFICACIÓN

La implementación de un SmartLab o laboratorio inteligente ofrecería una solución integral para abordar estas problemáticas, optimizando el uso de recursos y mejorando las condiciones operativas del laboratorio. Un sistema de monitoreo de aforo permitiría apagar los equipos cuando no estén siendo utilizados, reduciendo el consumo energético hasta en un 30-40%, como lo respalda un estudio realizado por la International Energy Agency (2017), donde se estima que la automatización puede reducir considerablemente el gasto energético en entornos de trabajo. Adicionalmente, el control de temperatura y humedad es crítico para la longevidad y el rendimiento de los servidores, ya que se estima que, por cada 10 grados de aumento en la temperatura ambiente, la vida útil de un servidor se reduce a la mitad. La integración de un sistema inteligente de regulación ambiental ayudaría a mantener las condiciones óptimas, asegurando un funcionamiento más eficiente y seguro. En conjunto, estas mejoras contribuirán no solo a un ahorro significativo en los costos operativos de la universidad, sino también a una reducción de la huella de carbono, alineándose con los objetivos de sostenibilidad y eficiencia energética.

LEVANTAMIENTO DE REQUERIMIENTOS

Para el levantamiento de requerimientos, se hizo una entrevista al profesor encargado del laboratorio Nicolás Segura el cual fue el que solicitó algún tipo de automatización dentro de la jaula de los servidores y el ahorro energético de los computadores.

IDENTIFICACIÓN DE LOS INTERESADOS DEL SISTEMAS

Interesado	Rol	Intereses	Impacto	Participación
Coordinadores del laboratorio (Asistentes Graduados)	Usuarios principales	Uso eficiente del laboratorio, disponibilidad de equipos	Alto	Alta
Estudiantes (pregrado, maestría y doctorado)	Usuarios secundarios	Acceso a equipos funcionales, ambiente confortable	Alto	Baja
Universidad (Administrativos del departamento de sistemas)	Usuario principal	Ahorro energético y mejor gestión de los recursos del departamento, mejores métricas en informes	Medio	Baja
Profesores del departamento de sistemas	Usuario secundario	Disponibilidad de equipos, integridad física en los recursos usados, Mayores resultados en sus actividades	Medio	Baja

ESPECIFICACIÓN Y PRIORIZACIÓN DE LOS REQUISITOS FUNCIONALES

Requerimientos funcionales para el apagado automático de los equipos

- **Identificar los equipos en uso:** El sistema es capaz de identificar qué equipos están apagados o prendidos dentro del laboratorio.
- **Apagado automático:** El sistema debe apagar los computadores si no detecta una presencia dentro de una distancia de un metro y un intervalo de 15 minutos.

Requerimientos funcionales para la detección y monitoreo de humedad en la jaula de racks

- **Monitoreo de Temperatura:** El sistema monitorea la temperatura dentro de la jaula de Racks.
- **Monitorio de la humedad:** El sistema monitorea la humedad dentro de la jaula de Racks.
- **Notificación en caso de emergencia:** El sistema al detectar condiciones no aptas (temperatura y humedad) para la jaula de racks genera una notificación al celular del coordinador del laboratorio del momento.
- **Regulación de temperatura del aire acondicionado:** Una vez se detecta un cambio que propicie un riesgo generar una regulación de temperatura con el aire acondicionado.

ESPECIFICACIÓN Y PRIORIZACIÓN DE LOS REQUISITOS DE CALIDAD

Requisitos de calidad para el apagado automático de los equipos

- **Precisión:** Se requiere que la detección de presencia sea precisa para no generar inconvenientes con el uso de los computadores.
- **Mantenibilidad:** El sistema debe ser de fácil mantenimiento como la instalación, implementación y actualización de los componentes a utilizar.
- **Usabilidad:** El sistema debe ser capaz de personalizar las variables inicialmente definidas y las alertas dependiendo de las necesidades del coordinador del laboratorio.

Requisitos de calidad para la detección y monitoreo de humedad en la jaula de servidores

- **Fiabilidad:** Se requiere que los dispositivos tengan una probabilidad baja de fallos para que el monitoreo sea constante.
- **Precisión:** Se necesita que los datos en la mayor parte del tiempo sean reales y precisos.
- **Disponibilidad:** El sistema debe estar disponible la mayor parte del tiempo ya que es un monitoreo en tiempo real y 24/7.
- **Usabilidad:** El sistema debe ser capaz de personalizar las variables inicialmente definidas y las alertas dependiendo de las necesidades del coordinador del laboratorio.

DISEÑO DEL SISTEMA A NIVEL DE HARDWARE DE SENSADO, ACTUACIÓN Y GATEWAY IOT

DETERMINAR Y CARACTERIZAR LAS VARIABLES DE MONITOREO

Variables de monitoreo para el apagado automático de los equipos

- **Estado de los computadores**: Un binario que indica si está encendido (1) o apagado (0).
- **Distancia de detección de presencia:** El rango de distancia medido en metros donde se puede decidir si apagar o no el equipo.
- **Tiempo transcurrido sin la presencia de un usuario:** Intervalo de tiempo en minutos para decidir si se puede o no apagar el computador.

Variables de monitoreo para la detección y monitoreo de humedad y temperatura en la jaula de racks

- **Temperatura:** Variable en °C para medir la temperatura de la jaula de racks.
- **Humedad:** Variable de % de humedad relativa del aire para el monitoreo dentro de la jaula de racks.

REALIZAR UN DIAGRAMA DE BLOQUES DEL FUNCIONAMIENTO PROPUESTO PARA LA CAPA DE DISPOSITIVOS DEL SISTEMA IoT

Diagrama de bloques para el apagado automático de computadores

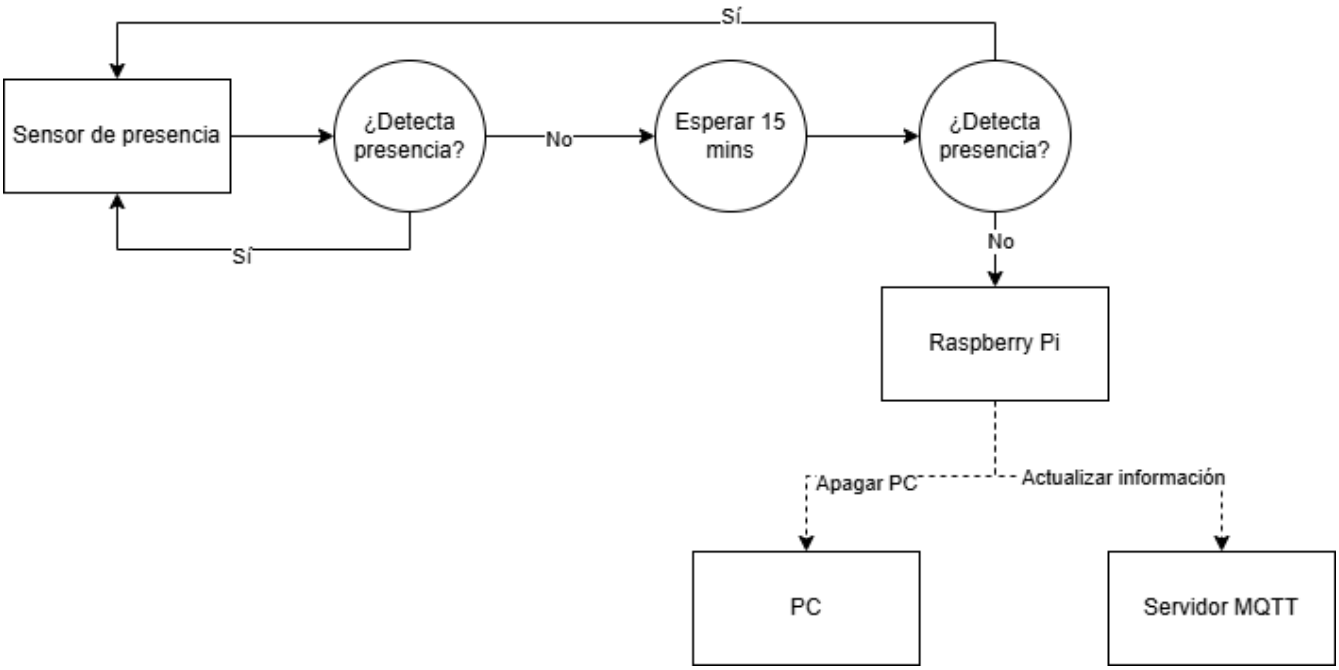
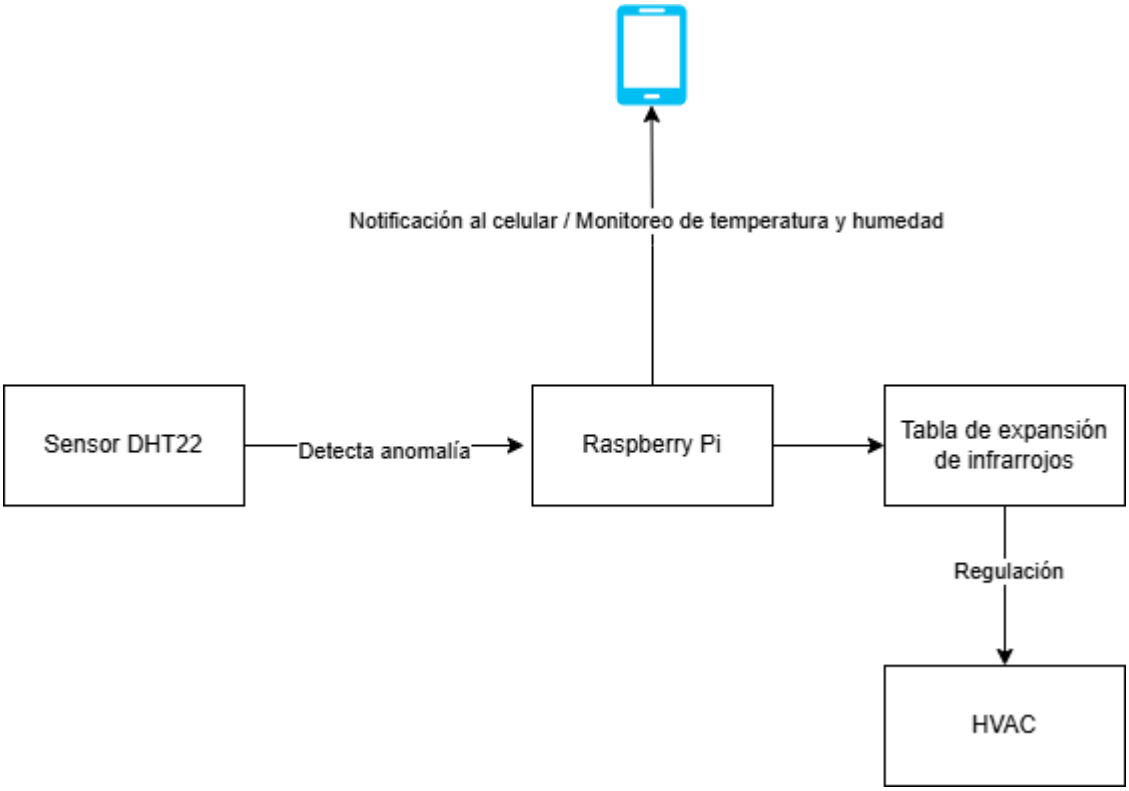


Diagrama de bloques para la detección de humedad y temperatura de la jaula de racks



SELECCIONAR LOS DISPOSITIVOS DE SENSADO A UTILIZAR

Sensores para el apagado automático de los equipos

- **Sensores de movimiento PIR (Passive Infrared Sensor)**: Los sensores PIR son ideales para detectar la presencia de personas en el laboratorio debido a su alta precisión en la detección de movimiento humano, con un rango de detección típico de 5 a 12 metros y un ángulo de detección de 110° a 180°. Además, tienen un rápido tiempo de respuesta en el rango de milisegundos, un muy bajo consumo de energía y son relativamente económicos, lo que permite su implementación a gran escala sin un costo prohibitivo.

  La elección de este sensor se hizo teniendo en cuenta variables como costos y fácil implementación y adaptación al gateway Raspberry Pi que se va a utilizar.

Sensor de temperatura y humedad en la jaula de racks

- **Sensores de temperatura y humedad DHT22**: Los sensores DHT22 son adecuados para monitorear las condiciones ambientales del laboratorio debido a su alta precisión en la medición de temperatura (±0.5°C) y humedad (±2-5% RH). Ofrecen un amplio rango de medición, desde -40°C a 80°C para temperatura y 0-100% RH para humedad, con un rápido tiempo de respuesta y actualizaciones de datos cada 2 segundos. Son moderadamente económicos, ofreciendo una buena relación costo-beneficio para aplicaciones de monitoreo ambiental

SELECCIONAR LOS DISPOSITIVOS DE ACTUACIÓN A UTILIZAR

Dispositivo de actuación para el apagado automático de los equipos

- **Raspberry Pi:** Este microprocesador funcionará como actuador remoto ya que se encargará del apagado de los equipos por medio de conexión remota hacia estos.

Dispositivo de actuación para la regulación de temperatura en la jaula de racks

- **Raspberry Pi/Tabla de expansión de infrarrojos**: Por medio de una tabla de expansión de infrarrojos junto con la raspberry pi se generarán comunicaciones a través de infrarrojo para la regulación de aire acondicionado del lugar dependiendo de la temperatura y humedad actual.

DETERMINAR LOS PROCESOS A REALIZAR EN EL GATEWAY IoT

- **Recopilación de datos**: El Gateway IoT recopilará datos de todos los sensores conectados, incluyendo sensores de movimiento, temperatura y humedad. Este proceso implica la lectura continua de los datos proporcionados por los sensores y su almacenamiento temporal en el dispositivo.

- **Preprocesamiento**: Antes de enviar los datos a la nube o a un servidor central, el Gateway IoT realizará un preprocesamiento de los datos. Esto incluye la filtración de datos ruidosos o irrelevantes, la normalización de los datos para asegurar consistencia y la agregación de datos para reducir la cantidad de información a transmitir.

- **Almacenamiento**: El Gateway IoT almacenará temporalmente los datos recopilados para asegurar que no se pierdan en caso de interrupciones en la conectividad. Este almacenamiento temporal también permite realizar análisis locales rápidos sin necesidad de enviar todos los datos a la nube, esto es útil principalmente para los datos de los sensores de proximidad que requieren de un procesamiento de borde para tomar decisiones acerca del apagado de los computadores.

- **Análisis**: El Gateway IoT puede realizar análisis preliminares de los datos para detectar patrones o eventos significativos. Por ejemplo, puede identificar si hay un aumento repentino en la temperatura o humedad, que podría indicar un problema con algún dispositivo externo al sistema y ser una señal de alerta.

- **Transmisión segura**: Una vez preprocesados y analizados, los datos se transmitirán de manera segura a la nube o a un servidor central. Esto implica el uso de protocolos de seguridad como MQTT con TLS para asegurar que los datos no sean interceptados o alterados durante la transmisión.

SELECCIONAR EL GATEWAY IoT A UTILIZAR

La Raspberry Pi 4, configurada como un gateway IoT, es una opción rentable y flexible que puede manejar la recopilación, procesamiento, y envío de datos de los sensores y actuadores a una nube o servidor local para su análisis y control. Combinada con software como Azure IoT Edge o AWS Greengrass, puede funcionar como un gateway potente para este proyecto.

Características Específicas

- **Conectividad**
  - WiFi y Ethernet: La Raspberry Pi 4 cuenta con conectividad WiFi 802.11ac y un puerto Ethernet Gigabit, permitiendo conectar los sensores que utilicen WiFi directamente al gateway o mediante un switch de red para los dispositivos Ethernet.
  - Bluetooth 5.0: Ideal para dispositivos de sensado como control de aforo o temperatura que utilicen comunicación Bluetooth.
  - Puertos USB y GPIO: Dispone de varios puertos USB y un conjunto de pines GPIO que permiten conectar dispositivos adicionales, como módulos Zigbee o Z-Wave, ampliando la compatibilidad con sensores que utilicen estos protocolos.

- **Capacidad de Procesamiento**
  - Procesador de cuatro núcleos y 4 GB o 8 GB de RAM, lo cual es suficiente para procesar datos en tiempo real y ejecutar software de monitoreo, permitiendo un análisis local preliminar de los datos antes de enviarlos a la nube.
  - Software IoT Edge (Azure IoT Edge o AWS Greengrass):
  - Azure IoT Edge o AWS Greengrass permiten la integración de la Raspberry Pi como un gateway inteligente que puede procesar los datos de los sensores y actuar según reglas predeterminadas sin depender completamente de la conexión a internet.
  - El software permite la ejecución de contenedores Docker, lo que facilita el procesamiento local, la toma de decisiones y la automatización, como apagar los equipos según la interpretación de la señal del sensor de movimiento infrarrojo.

- **Seguridad**
  - Soporte para cifrado de datos y la capacidad de implementar mecanismos de autenticación para el control de acceso, asegurando que solo los dispositivos autorizados puedan interactuar con el gateway.

- **Escalabilidad y Flexibilidad**
  - Capacidad para gestionar múltiples sensores y actuadores simultáneamente, y es fácilmente ampliable si se requieren más dispositivos en el futuro.

Integración con los Sensores y Actuadores:

- Los sensores de proximidad que utilizan WiFi o Bluetooth pueden conectarse directamente a la Raspberry Pi.

- Los sensores de temperatura y humedad que utilizan protocolos como I2C o SPI pueden conectarse a través de los pines GPIO.
- Los actuadores (como se mencionaron anteriormente) se pueden controlar directamente a través de los pines GPIO o mediante módulos de expansión conectados a los puertos USB.

La implementación de una Raspberry Pi 4 como gateway IoT es ampliamente respaldada por experiencias de proyectos de IoT en entornos educativos y empresariales, como lo menciona el informe de IoT For All en su artículo “Choosing the Right IoT Gateway” (2022), donde se destaca su flexibilidad y capacidad para funcionar como un gateway confiable y económico para diversos sistemas IoT.

Definición de la arquitectura de comunicación

Determinar los protocolos de la capa física y de enlace a utilizar en el Sistema IoT

Se decidió implementar Wi-Fi 802.11 tradicional (2,4 Ghz), debido a que es una elección estratégica tanto para sistemas de apagado automático de equipos como para la detección y monitoreo de humedad en jaulas de racks. Esta tecnología ofrece una serie de ventajas que la hacen especialmente adecuada para las aplicaciones de este proyecto.
En el contexto de un sistema de apagado automático de equipos, la banda de 2,4 GHz proporciona una cobertura amplia, lo que permite abarcar áreas extensas con un número reducido de puntos de acceso. Esto facilita la comunicación efectiva entre los sensores de presencia y los equipos, incluso en salas amplias de la universidad. Además, la infraestructura Wi-Fi de 2,4 GHz está ampliamente implementada y es compatible con una variedad de dispositivos, lo que simplifica su integración en sistemas existentes sin necesidad de equipos adicionales.
Por otro lado, en la detección y monitoreo de humedad en jaulas de racks, la banda de 2,4 GHz es capaz de soportar múltiples conexiones simultáneas, lo cual es esencial en entornos con numerosos dispositivos electrónicos. Esto asegura que los sensores de humedad y temperatura transmitan datos de manera confiable. Aunque otras tecnologías como Wi-Fi HaLow operan en frecuencias sub-1 GHz y ofrecen mejor penetración a través de obstáculos, la banda de 2,4 GHz también proporciona una penetración adecuada para la mayoría de las aplicaciones internas, facilitando la comunicación entre sensores en entornos complejos.
En resumen, la elección de Wi-Fi 802.11 en la banda de 2,4 GHz para estas aplicaciones se basa en su equilibrio entre cobertura, capacidad de manejo de múltiples conexiones y compatibilidad con infraestructuras existentes, lo que facilita su implementación y operación efectiva en

Determinar el protocolo de enrutamiento a utilizar en el Sistema IoT

En este proyecto las conexiones son de punto a punto, lo que significa que cada dispositivo se comunica directamente con el Gateway IoT sin necesidad de pasar por otros nodos intermedios. Esto elimina la complejidad asociada con las redes de malla, donde los datos deben ser enrutados a través de múltiples nodos para llegar a su destino. Al no tener una topología de red compleja, no es necesario implementar un protocolo de enrutamiento, lo que simplifica la arquitectura de la red y reduce los requisitos de configuración y mantenimiento. Esta simplicidad también mejora la eficiencia y la fiabilidad del sistema,

ya que cada dispositivo tiene una ruta de comunicación directa y clara, minimizando posibles puntos de fallo y optimizando el uso de recursos.

Determinar el protocolo de encapsulamiento a utilizar en el Sistema IoT

Galgus (2021) menciona que protocolos de encapsulamiento como 6LoWPAN "realiza compresión de encabezados y fragmentación para permitir la transmisión de paquetes IPv6 en redes de baja potencia con limitaciones en el tamaño de las tramas". No obstante, al utilizar Wi-Fi tradicional en entornos de laboratorio inteligente, se aprovecha el soporte nativo de IPv4 e IPv6 integrado en las pilas de red estándar de los dispositivos Wi-Fi modernos, tanto en puntos de acceso como en sensores compatibles. Esta compatibilidad elimina la necesidad de recurrir a protocolos adicionales destinados a la adaptación de IPv6 sobre medios restringidos, como 6LoWPAN, empleado principalmente en redes de baja potencia y bajo ancho de banda, típicas del Internet de las Cosas (IoT) en entornos con recursos muy limitados.
Los dispositivos Wi-Fi certificados bajo los estándares actuales (como IEEE 802.11n, 802.11ac o 802.11ax) cuentan con pilas de red que soportan IPv6 de manera nativa. De acuerdo con la Wi-Fi Alliance (2022), la interoperabilidad con IPv6 es un factor clave para el despliegue de servicios avanzados en redes domésticas, empresariales e industriales. Además, se hace referencia a la capacidad de Wi-Fi de interoperar con protocolos IP, incluyendo IPv6, sin requerir adaptaciones especiales. Finalmente, es preciso mencionar que en un entorno Wi-Fi, los dispositivos cuentan con mayor ancho de banda, mejor capacidad de proceso y no necesitan este tipo de adaptación. La convergencia directa de IPv6 sobre Wi-Fi simplifica la arquitectura, al no requerir gateways ni capas intermedias para la encapsulación o compresión de datos. Esta simplificación queda reflejada en artículos técnicos de la IETF, como el RFC 6282, donde se detalla la compresión de cabeceras IPv6 para 6LoWPAN, algo innecesario en entornos con Wi-Fi tradicional.
Al prescindir de protocolos adicionales, se reduce la complejidad administrativa y se minimizan los potenciales cuellos de botella. Esto se alinea con las necesidades operativas de un laboratorio inteligente, donde la fiabilidad, la baja latencia y la simplicidad de gestión son primordiales. Estudios y guías de fabricantes líderes como Cisco resaltan la importancia de una infraestructura de red IPv6 nativa para soportar de forma más eficiente la creciente cantidad de dispositivos conectados, así como para simplificar el troubleshooting y el monitoreo.

Determinar el protocolo de descubrimiento de servicios a utilizar en el Sistema IoT

Se decidió optar por el protocolo de descubrimiento de servicios mDNS, ya que es útil para el sistema IOT planteado para el laboratorio inteligente porque permite la detección automática de dispositivos dentro de una red local sin necesidad de un servidor DNS central, lo que disminuye costos y es suficiente para este contexto, pero, además, ofrece otras ventajas como una mayor agilidad en la integración de sensores y dispositivos. Al implementar mDNS, el gateway Raspberry Pi puede identificar y conectarse fácilmente con los sensores PIR y DHT22, y otros que se pueden agregar posteriormente, a medida que se integran al sistema, sin requerir configuraciones manuales en cada dispositivo. Además, este protocolo facilita el uso de nombres de red amigables en lugar de direcciones IP, simplificando el acceso y la administración de los dispositivos conectados. Si no se implementara mDNS, cada sensor tendría

que configurarse manualmente, lo que incrementaría la carga de trabajo y reduciría la escalabilidad del sistema, ya que cualquier expansión o cambio requeriría ajustes detallados y menos eficientes

Determinar el protocolo de la capa de aplicación a utilizar en el Sistema IoT

Aprovechando la capacidad de procesamiento y conectividad de una Raspberry Pi 4 como gateway del sistema IoT, se puede montar en el mismo dispositivo un servidor MQTT, consolidando así la función de enrutador y la de broker de mensajería en una sola plataforma. Esto simplifica la arquitectura general del sistema, ya que no se requiere un servidor externo para la gestión de la capa de aplicación. De esta manera, el flujo de datos entre los sensores (como los PIR para presencia, o los DHT22 para temperatura y humedad), el sistema de control y los dispositivos cliente se vuelve más ágil y fácil de mantener.

Al integrar la capa de aplicación a través del protocolo MQTT en la Raspberry Pi 4, el sistema puede gestionar el encendido y apagado de los computadores de forma eficiente. MQTT, con su modelo de publicación/suscripción, brinda un canal de comunicación ligero que optimiza el intercambio de mensajes entre sensores de presencia y el sistema de control. La información enviada por los sensores (por ejemplo, si un computador está encendido o apagado) se transmite de manera rápida y con un uso mínimo de ancho de banda. Esta eficiencia se traduce en una latencia muy baja, asegurando que los comandos de apagado lleguen a tiempo, evitando consumos eléctricos innecesarios y alargando la vida útil de los equipos.

La resiliencia de MQTT ante conexiones intermitentes resulta beneficiosa en el caso de fallos esporádicos de la red. Dado que el broker MQTT puede operar en la propia Raspberry Pi 4, se reduce la dependencia de servicios externos, manteniendo la comunicación y el control interno del laboratorio. Además, el soporte de autenticación por usuario y contraseña, junto con el cifrado de la información mediante TLS, garantiza que la comunicación de datos sensibles (como el estado de los equipos) se realice de forma segura, protegiendo la información y evitando accesos no autorizados. Del mismo modo, el protocolo MQTT es idóneo para el monitoreo de la humedad mediante sensores DHT22 instalados en la jaula donde se encuentran los equipos. El diseño ligero de MQTT facilita la transmisión de datos de humedad en tiempo real hacia el servidor (alojado en la misma Raspberry Pi 4), permitiendo detectar variaciones en las condiciones ambientales de inmediato. Ante cualquier cambio que ponga en riesgo la integridad de los equipos, como un aumento excesivo de la humedad, el sistema puede procesar rápidamente dicha información y emitir alertas o activar medidas correctivas, ya sea a través de notificaciones, ajustes de climatización o acciones preventivas.

La capacidad para manejar múltiples sensores sin consumir excesivos recursos hace de MQTT una herramienta escalable. A medida que el laboratorio crezca, se podrán agregar más sensores y nodos IoT sin comprometer el rendimiento ni la calidad del servicio. La publicación/suscripción facilita la expansión del sistema, ya que la Raspberry Pi 4 puede seguir actuando como el punto central de intercambio de información, procesando las mediciones, almacenándolas si fuese necesario, y enviando alarmas o comandos a los dispositivos finales.

Definición de la Integración Fog /Cloud

Determinar la integración de las capas de cloud y/o fog computing a la arquitectura del sistema propuesto

Nuestro diseño de la arquitectura propuesta para el SmartLab, se opta por una integración directa entre los dispositivos IoT y la nube, evitando la incorporación de una capa intermedia de Fog Computing. Aun cuando el Fog Computing resulta valioso en contextos donde la baja latencia y el procesamiento en tiempo real son críticos. Por ejemplo, sistemas de automatización industrial o entornos con conectividad inestable, en el SmartLab las necesidades operativas difieren.

Latencia y Tiempo Real

En el caso del laboratorio inteligente, la transmisión de datos de sensores de presencia (PIR), humedad (DHT22), y el envío de comandos de apagado o encendido de equipos no exige una respuesta inmediata a nivel de milisegundos. Por ejemplo, si un computador detecta inactividad, que su apagado se demore unos segundos en ejecutarse no compromete la funcionalidad del sistema. Del mismo modo, variaciones en la temperatura o humedad no requieren una reacción instantánea a nivel local; los datos pueden ser procesados en la nube sin afectar la eficacia de las acciones correctivas o preventivas. De esta manera, la baja latencia que normalmente justificaría el Fog Computing no aporta un beneficio significativo en el SmartLab.

Reducción de Costos y Complejidad

La inclusión de nodos Fog añade costos adicionales en hardware, software y mantenimiento. Estos nodos requieren infraestructura especializada, gestión de software intermedio y potencia de cómputo extra, complicando el escalamiento y el soporte técnico. En el contexto del SmartLab, donde la Raspberry Pi 4 funge como gateway para el sistema IoT y puede montar un servidor MQTT local, el preprocesamiento necesario ya se realiza de forma nativa en el dispositivo. Por ejemplo, la Raspberry Pi puede filtrar datos irrelevantes, agrupar lecturas y aplicar reglas simples antes de enviarlas a la nube. Así se evita la necesidad de una capa intermedia de Fog que no agregaría un valor proporcional al incremento de complejidad.

Entorno Controlado y Estable

El SmartLab opera en un entorno con conectividad estable y predecible. A diferencia de entornos industriales expuestos a variaciones abruptas en la red, la infraestructura del laboratorio asegura un enlace confiable con la nube. En este escenario, la función principal del gateway, ya sea la Raspberry Pi 4 u otro dispositivo, consiste en agregar, estructurar y, cuando sea preciso, encriptar las transmisiones hacia el servidor en la nube (donde se aloja la capa de aplicación y el procesamiento más complejo de datos). Por eso mismo, no es imprescindible una capa Fog para evitar interrupciones o garantizar resiliencia; el enlace se mantiene lo suficientemente robusto como para transmitir datos sin pausas críticas que pongan en riesgo la operación.

Integración Directa con la Nube

El resultado es una arquitectura donde el IoT Gateway (la Raspberry Pi 4) actúa como punto central de recepción y preprocesamiento básico de datos. Desde allí, a través del protocolo MQTT, se envían los datos a la nube para su almacenamiento, análisis y toma de decisiones. Esta nube puede alojar herramientas analíticas, bases de datos, dashboards y sistemas de inteligencia artificial, que aprovechan las características escalables y centralizadas del Cloud Computing.

Describir las funciones de las capas de cloud y/o fog computing en el sistema propuesto, de haber sido consideradas

Capa Fog Computing

- **Preprocesamiento Local de Datos: l**a capa Fog se ubicaría entre los dispositivos IoT (sensores y actuadores) y la nube, ejerciendo como un punto intermedio de cómputo. Aquí se podrían realizar tareas sencillas de filtrado, agregación o normalización de datos antes de enviarlos a la nube, reduciendo el volumen de información que viaja por la red.
- **Reducción de Latencia**: en sistemas donde la respuesta en tiempo real resulta crítica, el Fog podría ejecutar lógicas de control local, desencadenando acciones inmediatas sin la necesidad de esperar una respuesta desde la nube. Por ejemplo, un cambio súbito en las condiciones ambientales detectado por un sensor podría activar una reacción local sin retrasos.
- **Resiliencia y Disponibilidad:** ante interrupciones temporales en la conexión a la nube, la capa Fog podría continuar operando y tomando decisiones autónomas a corto plazo. Esto asegura un nivel mínimo de funcionalidad aunque la infraestructura en la nube no esté disponible de forma temporal.
- **Filtrado de Seguridad y Privacidad:** el Fog podría implementar protocolos de seguridad locales, como cifrado inicial o anonimización de datos, antes de compartir información con el servidor en la nube, añadiendo una capa de protección y cumpliendo requerimientos normativos.

Capa Cloud Computing

- **Análisis Avanzado y Almacenamiento Masivo:** la nube actuaría como el repositorio central donde se almacenan grandes volúmenes de datos históricos recolectados por los sensores del laboratorio. Aquí, se podría aplicar analítica avanzada, ya sea a través de herramientas de Big Data, Machine Learning o Inteligencia Artificial, para detectar patrones, tendencias y comportamientos anómalos.
- **Gestión Centralizada y Escalabilidad:** la capa Cloud proporcionaría la capacidad de escalado dinámico, permitiendo agregar más sensores, más usuarios o nuevas funciones sin grandes modificaciones en la infraestructura física. Además, el control y la administración del sistema (gestión de usuarios, configuración de dispositivos, políticas de seguridad) se llevarían a cabo desde una ubicación centralizada.
- **Integración con Aplicaciones Externas: l**a nube serviría como punto de conexión con aplicaciones de terceros, servicios externos o plataformas de análisis adicionales. Por ejemplo,

integrar el sistema con soluciones de gestión energética, monitoreo ambiental externo o sistemas de alerta basados en mensajería móvil.

- **Seguridad a Nivel Global:** la nube puede ofrecer servicios de autenticación más robustos, protección ante ataques cibernéticos y cumplimiento de normativas internacionales. Al centralizar la seguridad, se refuerzan controles como la autenticación multifactor, la detección de intrusiones y la verificación de integridad de los datos.

Definición de los Requerimientos de Seguridad

Requisitos de seguridad a nivel de información

Confidencialidad

Para asegurar que la información que viaja entre los sensores (como PIR, DHT22), los gateways (por ejemplo, una Raspberry Pi 4 actuando como punto central) y la nube esté resguardada de terceros, se empleará encriptación de extremo a extremo. La comunicación MQTT y las API de consulta a la nube se asegurarán con el protocolo TLS (Transport Layer Security). Esta capa de cifrado evita que un atacante, incluso si accede a las transmisiones, pueda leer o manipular el contenido de las mismas. De esta forma, tanto datos sensibles (como estados de equipos, lecturas ambientales o configuraciones del sistema) como credenciales de autenticación estarán protegidos.

Integridad

Para preservar la exactitud de la información, se integrarán algoritmos hash (como SHA-256) o funciones de integridad equivalentes en los mensajes transmitidos. Estos hash se enviarán junto con las cargas útiles para verificar que los datos recibidos no hayan sido alterados intencionalmente o producto de fallos en la comunicación. Finalmente, al validarse el hash en el destino (ya sea en la nube o en el gateway), se garantiza que los datos sean fidedignos, facilitando la toma de decisiones y la generación de alertas fiables.

requisitos de seguridad a nivel de acceso

Autenticación

Todos los usuarios, ya sean estudiantes, técnicos o administradores, y dispositivos deberán pasar por un proceso de autenticación basado en credenciales únicas (nombre de usuario/contraseña, certificados digitales o incluso autenticación de dos factores si es requerido). Esta medida impedirá accesos no autorizados a la plataforma, reduciendo el riesgo de que intrusos puedan manipular equipos o visualizar información confidencial.

Autorización

La definición de roles y perfiles de acceso es esencial. Por ejemplo, un estudiante podría únicamente visualizar el estado de los equipos y los valores ambientales, mientras que un técnico podría tener permisos para modificar la configuración de sensores, y un administrador podría tener control total (incluyendo la administración de usuarios). De esta manera, se segmenta el control, minimizando el

impacto de una cuenta comprometida y asegurando el principio de privilegio mínimo: cada actor tiene únicamente el nivel de acceso que necesita.

Control de Acceso Físico y Lógico

Se mantendrá una lista blanca (whitelist) de dispositivos autorizados, de modo que únicamente equipos reconocidos (por sus direcciones MAC, certificados, o claves preestablecidas) puedan conectarse a la red del laboratorio. En consecuencia, esta barrera previene que dispositivos no autorizados (por ejemplo, sensores falsos o puntos de acceso no legítimos) puedan inmiscuirse en la topología del laboratorio, garantizando un entorno de ejecución seguro.

requisitos de seguridad a nivel funcional

Disponibilidad

La disponibilidad es primordial para que el laboratorio funcione sin interrupciones y los sensores puedan reportar en tiempo real los cambios en las condiciones ambientales, que es lo más clave. Se implementarán mecanismos de redundancia en los gateways, como la existencia de un gateway secundario que pueda asumir las tareas del principal en caso de fallo. También se pueden planificar políticas de respaldo de energía o enlaces alternativos que mantengan la conectividad hacia la nube y entre los sensores y el gateway si el enlace principal se ve afectado.

Resiliencia

La resiliencia implica la capacidad del sistema para recuperarse de eventos inesperados, como la caída de un sensor crítico o la desconexión temporal de un gateway. Esto se logrará mediante herramientas de autosupervisión que detecten la ausencia de datos o la falla de componentes y reactiven dichos elementos, restaurando el servicio sin intervención manual. Por ejemplo, si un sensor deja de reportar, el sistema puede enviar una alerta a los administradores y, simultáneamente, poner en marcha un protocolo de sustitución de datos (por ejemplo, estimar valores temporales basados en lecturas previas).

Presentación de la Prueba de Concepto del Sistema IoT propuesto

Definir el esquema de prueba de concepto a utilizar

Objetivo de la Prueba de Concepto

Esta prueba tiene como objetivo emular la operación del SmartLab en un entorno controlado, con el fin de evaluar y demostrar las capacidades del sistema antes de su implementación física. A través de la simulación en CupCarbon, se busca:

- Validar la comunicación entre sensores (por ejemplo, sensores de temperatura, humedad y proximidad) y el gateway.
- Verificar la lógica de filtrado y preprocesamiento de datos en el Gateway.
- Confirmar el correcto funcionamiento de la capa de aplicación, representada en el gateway, que actúa como intermediario entre la red física (sensores) y la nube, siguiendo el modelo de publicación/suscripción (MQTT).

- Evaluar mecanismos de seguridad básicos, tales como el control de acceso basado en los IDs de dispositivos antes de procesar mensajes.
- Analizar el comportamiento del sistema ante diferentes condiciones ambientales (cambios de temperatura y humedad) y de ocupación (detección o ausencia del usuario).
- Asegurar la respuesta del sistema ante eventos clave, como la activación o desactivación automática de equipos según presencia del estudiante.

Entorno de Simulación en CupCarbon

1. **Representación Geográfica**
   La simulación se lleva a cabo sobre un mapa real (la imagen muestra un entorno de campus universitario), situando los nodos sensores y el gateway en la ubicación aproximada del laboratorio de redes. Esta representación geográfica permite observar la cobertura y el alcance de las señales inalámbricas y la topología de la red a una escala un poco grande debido a limitaciones del programa, se estima que para los sensores de proximidad se manejan las distancias 10:1.
2. **Nodos en el Escenario**
   - **Nodos Sensor (Temperatura y Humedad):** estos nodos envían lecturas periódicas al gateway. En el código del sensor se aprecia cómo, tras leer un valor (por ejemplo, la humedad en un rango), se genera una alerta si la condición se encuentra fuera de parámetros aceptables (por debajo de 40% o por encima de 60% en el caso de humedad, o mayor a 39°C para temperatura). Estas alertas se transmiten con un tipo de mensaje definido (“alertaHumedad” o “alertaTemp”).

     - **Script sensor humedad**

```
atget id id

loop
    delay 1000
    areadsensor humSen
    rdata humSen SensTipo idSens hum
    int humedad hum
    if ((humedad < 40)||(humedad > 60))
        data mens "alertaHumedad" id
        send mens 1
    end
```

     - **Script sensor temperatura**

```
atget id id

loop
    delay 1000
    areadsensor tempSen
    rdata tempSen SensTipo idSens te
    int temp te
    if (temp>39)
        data mens "alertaTemp" id
        send mens 1
    end
```

- Nodos Sensor de Proximidad/Presencia: representan equipos de cómputo y/o estaciones de trabajo. Estos nodos evalúan la distancia entre ellos y el nodo del "estudiante". Cuando el estudiante se acerca a una distancia predefinida, se considera que el equipo está en uso ("encendido"), y cuando se aleja por un tiempo mayor al umbral establecido, el sistema interpreta ausencia, enviando un mensaje al gateway para solicitar el apagado automático.

    - **Script sensores de proximidad**

```
atget id id
set id_estudiante 14
set distancia_deteccion 10
int t_sin_usuario 0
int t_con_usuario 0
set estado_pc "encendido"
loop
    read mens
    rdata mens tipo Nid
    if ((tipo == "apagar_pc_actuador") && (Nid == 1))
       set estado_pc "apagado"
       set t_sin_usuario 0
       set t_con_usuario 0
    end
    if ((tipo == "encender_pc") && (Nid == id_estudiante) && estado_pc == "apagado")
       set estado_pc "encendido"
       cprint "El pc fue encendido"
       set t_sin_usuario 0
    end
    if(estado_pc == "encendido")
      distance distancia id_estudiante
      if (distancia < distancia_deteccion)
         inc t_con_usuario
         set t_sin_usuario 0
         cprint "El estudiante esta usando el pc " id
      else
         inc t_sin_usuario
         set t_con_usuario 0
         cprint "El pc " id " no se esta usando"
         data mens "estudianteFuera" id t_sin_usuario
         send mens 1
      end
    end
    wait 1000
```

- Nodo Estudiante (Móvil): Un nodo que simula el desplazamiento del usuario real en el laboratorio. A medida que se mueve, su proximidad es registrada por los sensores de

presencia, lo que a su vez provoca la publicación de mensajes que permiten “encender” o “apagar” equipos según el patrón de uso.

- **Script sensor que representa estudiante**

```
atget id id
set distancia_deteccion 10
set id_pc_estudiante 7
set x 0
loop
    read mens
    rdata mens tipo valor
    distance distancia id_pc_estudiante
    if ((distancia < distancia_deteccion) && (x==0))
       data mens "encender_pc" id
       send mens id_pc_estudiante
       set x 1
    end
    delay 10000
    wait 100
```

- Nodo Gateway (Sink): Este nodo central actúa como el gateway IoT, recibiendo todos los mensajes de alerta o eventos provenientes de los sensores. Además, ejecuta lógica de decisión: ante una “alertaHumedad” o “alertaTemp”, el gateway imprime en la consola simulada la acción tomada (por ejemplo, “El deshumidificador se graduó usando el sensor infrarrojo” o “El aire acondicionado se graduó…”), reflejando la respuesta automática a condiciones ambientales adversas. Por otro lado, si recibe una alerta tipo “estudianteFuera” con un valor de tiempo superior a un umbral definido (2 segundos), envía un comando “apagar_pc_actuador” al sensor correspondiente, simulando el apagado remoto del equipo.

  - **Script del Gateway**

```
atget id id
set tiempo_apagado 2

set s_humedad_id 13
set s_temp_id 10
loop
    read mens
    rdata mens tipo Nid valor
    if ((tipo == "alertaHumedad") && (Nid == s_humedad_id))
        cprint "El deshumidificador se graduo usando el sensor infrarrojo"
    end
    if ((tipo == "alertaTemp") && (Nid == s_temp_id))
        cprint "El aire acondicionado se graduo usando el sensor infrarrojo"
    end
    if ((tipo == "estudianteFuera") && (valor > tiempo_apagado))
        data mens "apagar_pc_actuador" id
        send mens Nid
        cprint "El Pc " Nid " fue apagado"
    end

    wait 100
```

Mecanismos de Seguridad y Autenticación Simples

Aunque la simulación no despliega un entorno criptográfico complejo, sí incorpora elementos básicos de control de acceso y filtrado de mensajes:

- **Verificación de IDs:** en los scripts de los sensores y del gateway, se utilizan variables que almacenan el ID del dispositivo emisor y receptor. Antes de ejecutar acciones, el código verifica que el mensaje recibido provenga de un ID autorizado o esperado. Esta práctica simula una forma sencilla de control de acceso lógico, evitando que nodos no reconocidos puedan intervenir en el flujo de datos.
- **Modelado del Comportamiento del Gateway como Broker MQTT:** El gateway asume el rol equivalente a un servidor MQTT. En la lógica de los scripts, se ha manejado la "suscripción" y "publicación" mediante el envío y recepción de mensajes con tipos específicos ("alertaHumedad", "alertaTemp", "encender_pc", "apagar_pc_actuador"). Si bien no se está usando un broker MQTT real en la simulación, esta estructura refleja cómo ocurriría la interacción en un despliegue real, en el cual el gateway fungiría como punto central de intercambio, autenticación y filtrado de datos.

Validación de Requerimientos Funcionales y no Funcionales

Requerimientos Funcionales:

- **Monitoreo Ambiental (Temperatura y Humedad):** los sensores captan datos cada cierto intervalo (ej. cada 1 segundo), detectando condiciones fuera de rango. El gateway recibe estas alertas y "toma decisiones" impresas en la consola.
- **Detección de Presencia y Apagado Automático:** el sensor de proximidad asocia el estado del PC con la presencia del estudiante. Si el estudiante se retira y pasa un tiempo determinado, el gateway envía la orden de apagar. Esto comprueba la funcionalidad deseada para reducir el consumo energético.

- **Comunicación Asíncrona y Eficiente (MQTT Simulado):** los nodos publican mensajes cuando detectan un evento, y el gateway responde conforme a las reglas establecidas. Se consigue así el modelo pub/sub simplificado dentro de la simulación.

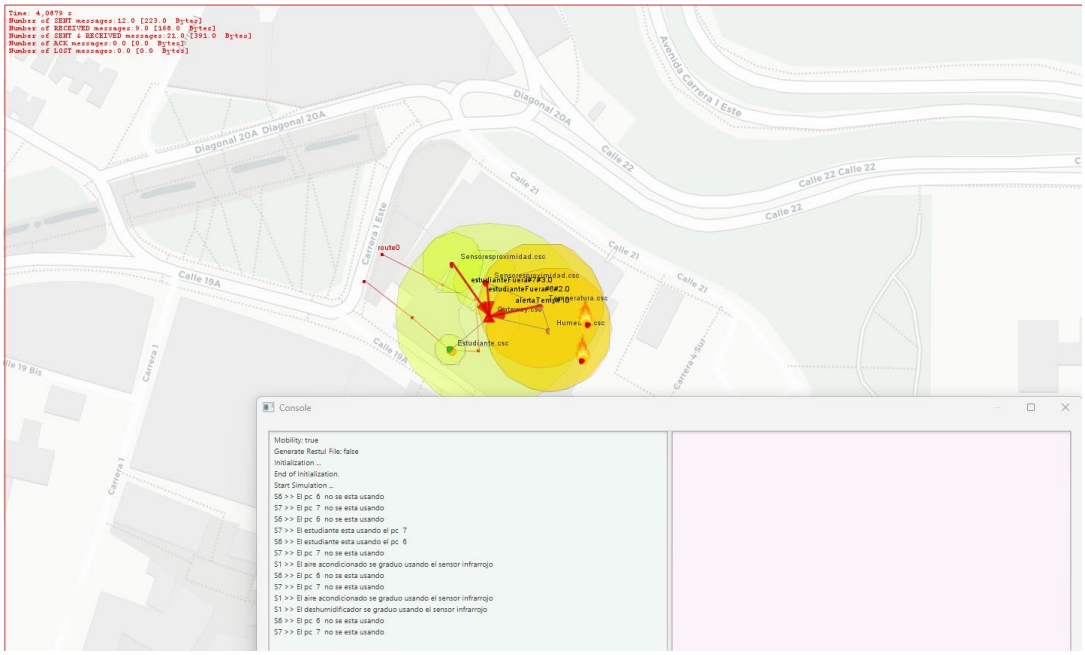
Requerimientos No Funcionales:

- **Latencia Adecuada:** aunque la simulación no mide tiempos reales de transmisión, la interacción continua y sin interrupciones entre nodos y gateway indica que, bajo el modelo hipotético, la latencia es lo suficientemente baja para las tareas encomendadas (no críticas).
- **Escalabilidad:** la simulación se puede extender fácilmente agregando más nodos sensor o más nodos actuador. El esquema MQTT simulado es flexible para alojar múltiples subscriptores y publicadores.
- **Seguridad Básica:** la verificación por ID ejemplifica un control de acceso básico, un primer paso hacia la integración de autenticación robusta, cifrado y roles de acceso, tal como se mencionó en las secciones de seguridad del documento.

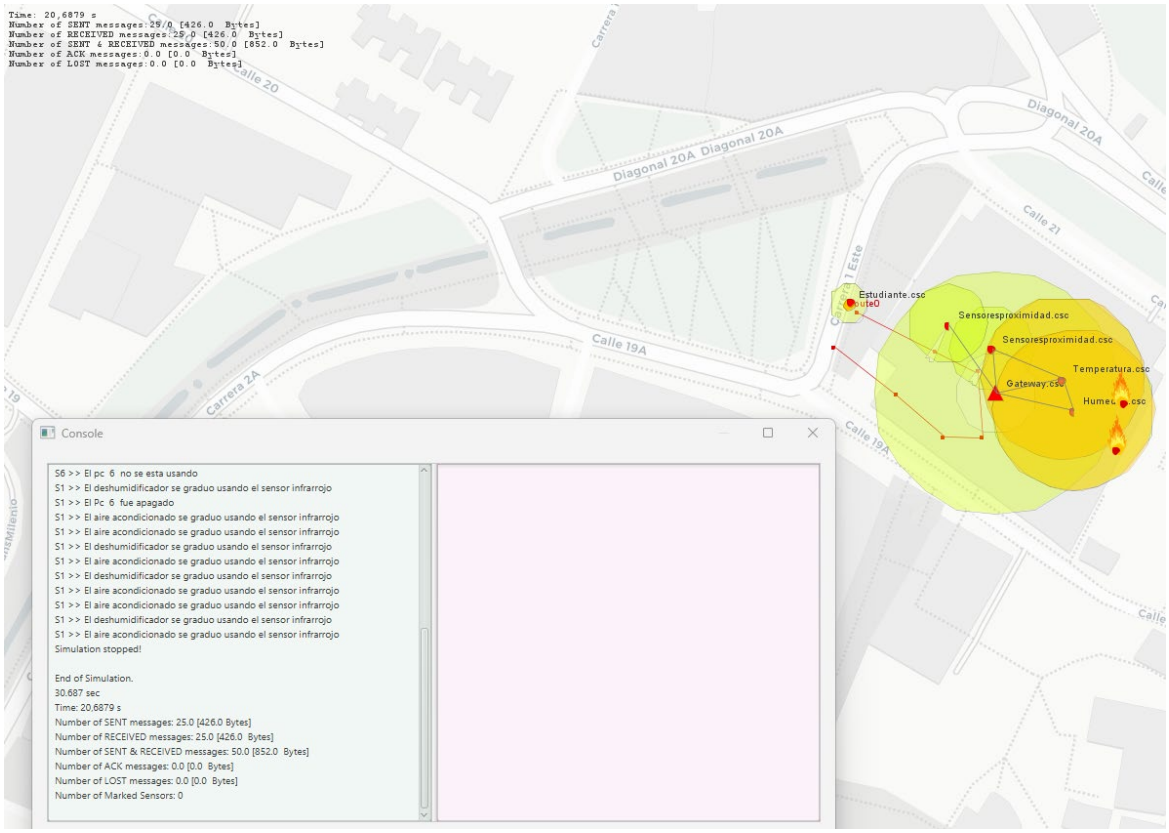
Documentar la prueba realizada y los resultados obtenidos

En las siguientes imágenes se puede observar el desarrollo de la simulación en CupCarbon:

- Mitad de recorrido



- Recorrido finalizado



Datos en la Esquina Superior Izquierda (Texto en Rojo)

- **Time (Tiempo):** Muestra el tiempo transcurrido durante la simulación. En la segunda captura, la simulación alcanzó los 20.6879 segundos.
- **Mensajes Enviados y Recibidos:**
  - Number of SENT messages: La cantidad total de mensajes enviados es 25.0 (426.0 Bytes).
  - Number of RECEIVED messages: La cantidad total de mensajes recibidos coincide con la cantidad enviada, lo que indica que no hubo pérdida de datos en la comunicación.
  - Number of LOST messages: El valor 0.0 (0.0 Bytes) confirma que no se perdieron mensajes durante la simulación.
  - Number of ACK messages: Se observa que no se configuraron mensajes de acuse de recibo (ACK), ya que el valor es 0.0.
- **Cobertura de los Nodos:** Las áreas circulares alrededor de los nodos representan el rango de alcance inalámbrico, mostrando una distribución adecuada de los sensores en el espacio.

Mensajes en la Consola (Parte Inferior de las Capturas)

- **Detección de Uso de PCs:**
  - "El PC 6 no se está usando" y "El PC 7 no se está usando" indican que los sensores de proximidad no detectan al estudiante dentro del rango.
  - "El estudiante está usando el PC 7" se registra cuando el nodo móvil (estudiante) entra en el área de detección de un sensor asociado al PC.

  - "El PC 6 fue apagado" refleja que, tras un periodo de ausencia del estudiante(2 segundos que simulan 20 minutos en la vida real), el sistema emite un comando desde el gateway para apagar el equipo.
- **Monitoreo de Condiciones Ambientales:**
  - Mensajes como "El aire acondicionado se graduó usando el sensor infrarrojo" y "El deshumidificador se graduó usando el sensor infrarrojo" muestran que el sistema responde a alertas de temperatura y humedad generadas por los sensores ambientales. Además, se observa como los sensores reportan al Gateway cuando detectan valores anómalos de medición.
- **Cierre de la Simulación:** La consola registra el final de la simulación, confirmando que no hubo errores ni eventos inesperados. Cabe resaltar que la simulación sigue corriendo con la iteración de mediciones de los sensores Gas que son cíclicas, pero se puede terminar sin problema la simulación.
-

Resultados Obtenidos

Validación de Requerimientos Funcionales:

- **Monitoreo Ambiental:** los sensores de temperatura y humedad detectaron condiciones fuera del rango y enviaron alertas que activaron respuestas automáticas (ajuste de aire acondicionado y deshumidificador).
- **Detección de Presencia y Uso de Recursos:** los sensores de proximidad asociaron la presencia del estudiante al estado de los PCs, activándolos o apagándolos según las condiciones definidas.
- **Comunicación entre Nodos:** el intercambio de mensajes fue exitoso, sin pérdidas ni errores de transmisión.

Validación de Requerimientos No Funcionales:

- **Latencia**: La respuesta del sistema fue inmediata, adecuada para las necesidades del SmartLab.
- **Confiabilidad**: No se perdieron mensajes durante la simulación, demostrando una alta fiabilidad en la comunicación.
- **Seguridad Básica**: Aunque el mecanismo de validación de IDs no es explícito en los resultados, el sistema operó bajo las reglas predefinidas, lo que implica que los nodos no autorizados no generaron interferencias.

Conclusión

La prueba realizada demuestra que el diseño del SmartLab es funcional y eficiente, cumpliendo con los objetivos planteados. El sistema:

- Responde adecuadamente a las condiciones ambientales y de ocupación.
- Gestiona los recursos energéticos (encendido y apagado de PCs) de manera automática.
- Mantiene una comunicación confiable y sin pérdidas.

Esta simulación ofrece una base sólida para la implementación física del sistema, con la posibilidad de agregar capas adicionales de seguridad y optimización para entornos reales.

BIBLIOGRAFÍA

- IEEE (1990). *IEEE Standard Glossary of Software Engineering Terminology* (IEEE Std 610.12-1990). Institute of Electrical and Electronics Engineers.
- Project Management Institute. (2017). *A guide to the project management body of knowledge (PMBOK guide)* (6th ed.). Project Management Institute.
- Raspberry Pi Foundation (s.f.). *Raspberry Pi Documentation*. Retrieved from https://www.raspberrypi.org/documentation/
- Arduino. (n.d.). *Arduino Documentation*. Retrieved from https://www.arduino.cc/en/Guide/HomePage
- Universidad de los Andes (2023). *Reporte Sostenibilidad 2023.* Recuperado de https://sostenibilidad.uniandes.edu.co/images/Reporte2023/REPORTE-2023_pag-web_compressed.pdf
- International Energy Agency (2017). *Digitalization and Energy*. Recuperado de https://www.iea.org/reports/digitalisation-and-energy
- ASHRAE Technical Committee (2015). *Thermal Guidelines for Data Processing Environments*. ASHRAE. Recuperado de https://xp20.ashrae.org/datacom1_4th/ReferenceCard.pdf.
- IoT For All (2022). Choosing the Right IoT Gateway. Recuperado de https://iot-solution.com/iotblog/_b0116.html
- Galgus. (2021). *Wi-Fi HaLow: La tecnología ideal para IoT*. Galgus. Recuperado de https://www.galgus.ai/blog/wifi-halow/
- Wi-Fi Alliance (2022). *Wi-Fi CERTIFIED HaLow highlights*. Wi-Fi Alliance. Recuperado de https://www.wi-fi.org/system/files/Wi-Fi_CERTIFIED_HaLow_Highlights.pdf