ESN-PdM Framework:

A TinyML-Driven IoT System for Condition Monitoring in Non-Stationary Mining Machinery

13/11 2024

Raúl de la Fuente

Motivación:
Sector Minero

Industria Minera

Para el año 2026, se proyecta que la industria minera global alcanzará un valor de mercado de $3.36 billones [1]

↘

Importancia

Sectores industriales dependen de la minería:

- Vehículos eléctricos
- Energías renovables
- Electrónica de consumo

Maquinaria

Actividades mineras dependen de:

- Camiones de acarreo
- Grúa horquilla
- Cintas transportadoras

Industria Minera

Condiciones Adversas

Maquinaria se encuentra sujeta a un entorno **hostil** y **altamente variable**.

- Humedad
- Impactos recurrentes
- Cargas pesadas
- Altas temperaturas
- Terreno irregular y/o accidentado
- Exposición al polvo

La exposición prolongada a estas condiciones conduce a la degradación del equipo, riesgos de seguridad e incluso detención de las operaciones [2].

Mantenimiento

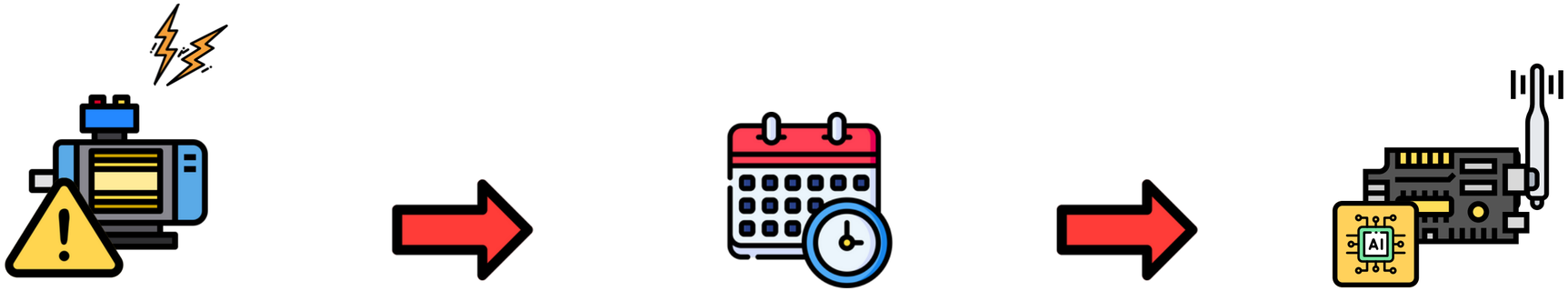
Estrategias de Mantenimiento

El mantenimiento es esencial para los sistemas de producción, garantizando el funcionamiento continuo, la seguridad y la durabilidad de los equipos [3].

↘

Evolución

Las estrategias de mantenimiento se han perfeccionado progresivamente desde el **correctivo**, pasando por el **preventivo** y culminando en el **predictivo**.



Corrective Maintenance

Preventive Maintenance

Predictive Maintenance

Mantenimiento Predictivo

*El mantenimiento predictivo (PdM) busca **anticipar fallas** en los sistemas a través de modelos matemáticos/físicos (model-driven) o análisis de datos (data-driven) [4].*

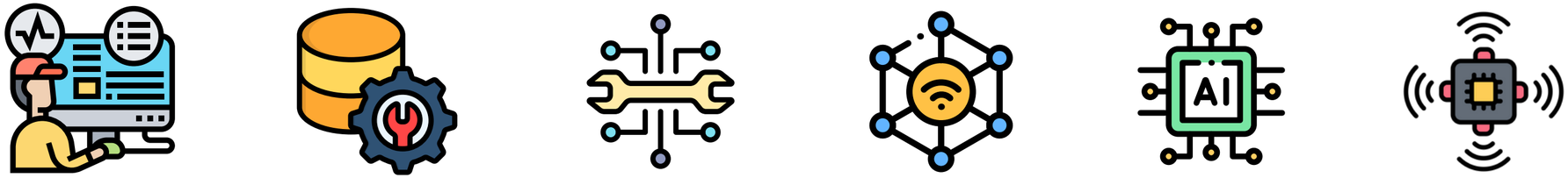
↘

Model-driven

- Enfoque “tradicional”
- Requiere conocimiento técnico.
- Costoso de implementar.

Data-driven PdM

- Actualmente el más popular.
- Fácil de implementar.
- *Data Acquisition y Data Processing.*



PdM:
Data Acquisition

Internet of Things

El Internet de las Cosas (IoT) busca interconectar objetos físicos a través de redes computacionales, permitiéndoles enviar y recibir datos [5].

↘

Áreas Involucradas

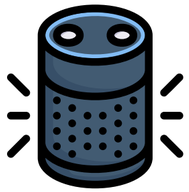
- Sistemas Embebidos
- Redes Computacionales
- Big Data
- HCI
- Ciberseguridad

Aplicaciones

- Smart watches
- Vehiculos autonomos
- Home assistants
- Sistemas domoticos
- Smart surveillance
- Control de flotas

- **Automatización Industrial (IIoT)**







Wireless Sensor Networks

Las Wireless Sensor Networks (WSNs) son redes IoT específicas con múltiples sensores que recopilan y transmiten datos en tiempo real de forma inalámbrica [6].

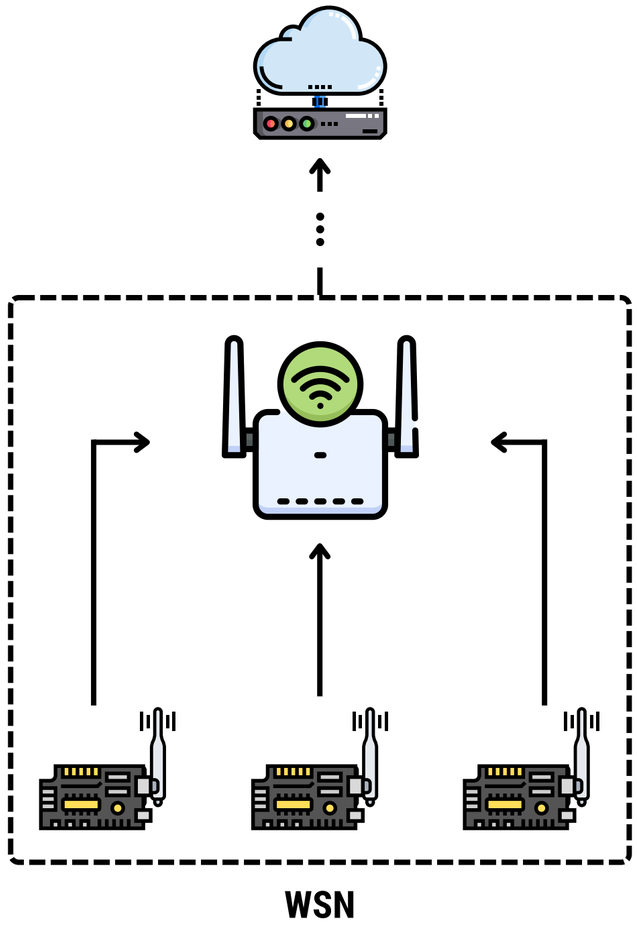
↘

Nodos

- Recopilan datos del entorno y los transmiten de forma inalámbrica a los gateways.
- *Microcontroller units* (MCUs) equipados con módulos de sensores.

Gateways

- Reciben la información de los nodos y la envían a la nube para su procesamiento y análisis.
- *Single-board Computers* (SBCs), MCUs o hardware dedicado.

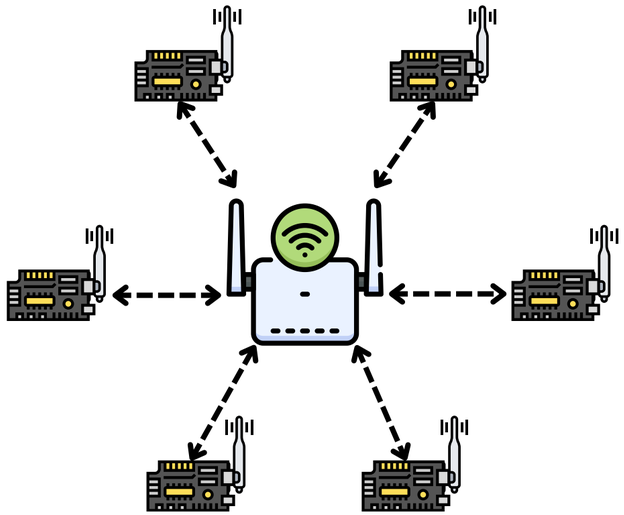


Wireless Sensor Networks

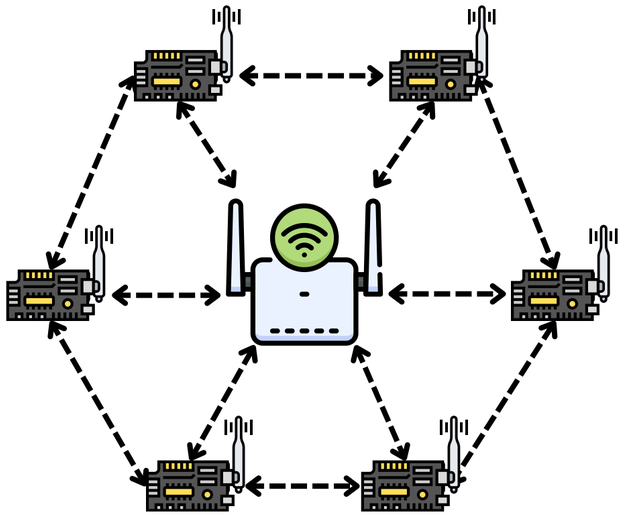
↘

Topologías

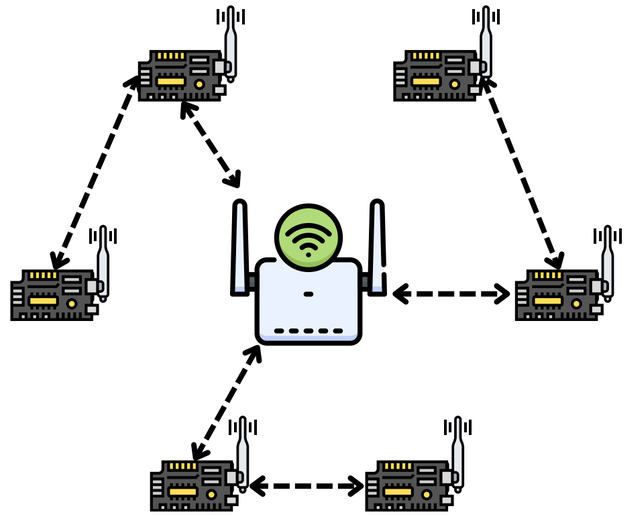
La topología de una WSN se refiere a cómo se organizan sus componentes, siendo las tres más comunes la **estrella**, la **malla** y la **híbrida**.



Star



Mesh



Hybrid

Wireless Sensor Networks

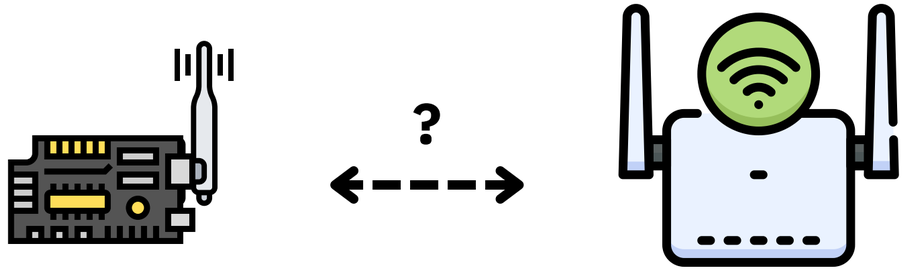
↘

Comunicación

En una WSN, los nodos se comunican de manera inalámbrica utilizando **pilas de protocolos estandarizadas**.

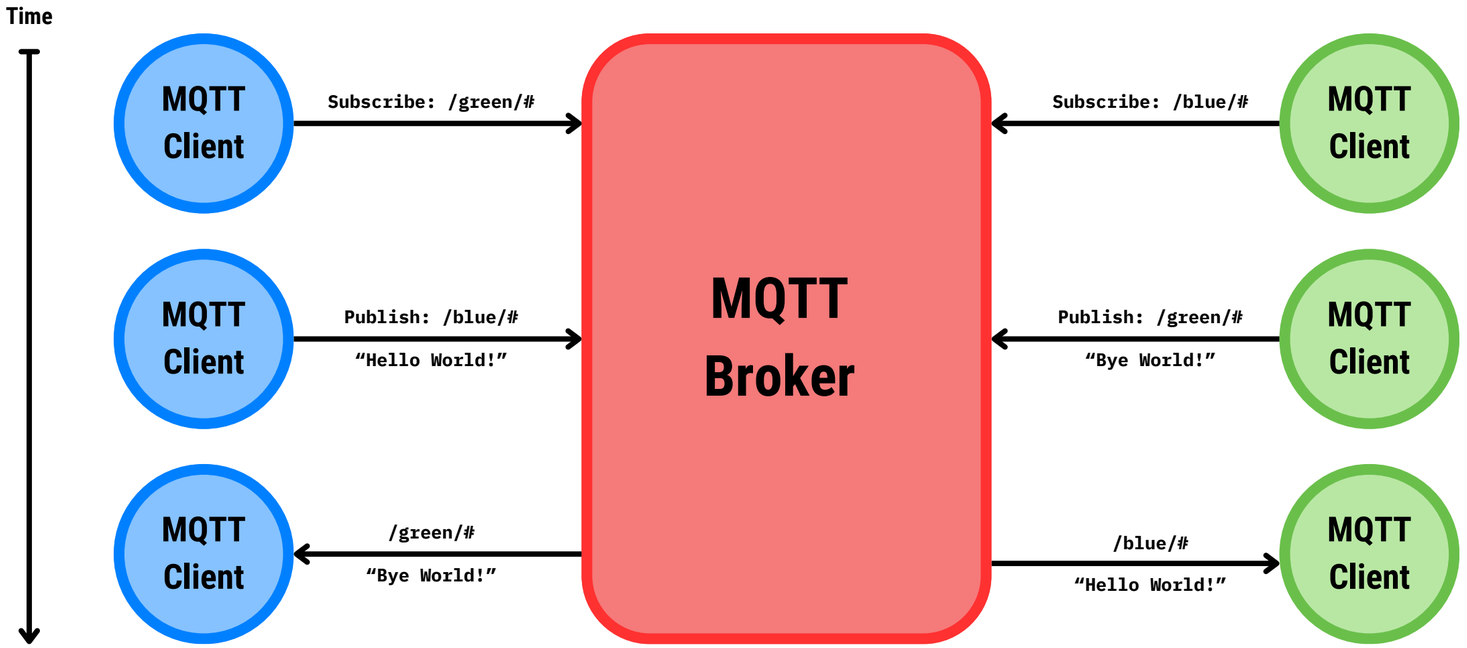
Algunos ejemplos son:

- MQTT (TCP/IP)
- HTTP (TCP/IP)
- ZigBee
- Bluetooth Low Energy (BLE)
- Matter (TCP/IP o Thread)

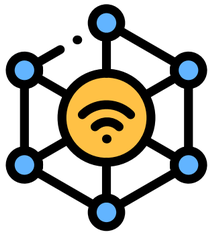


MQTT

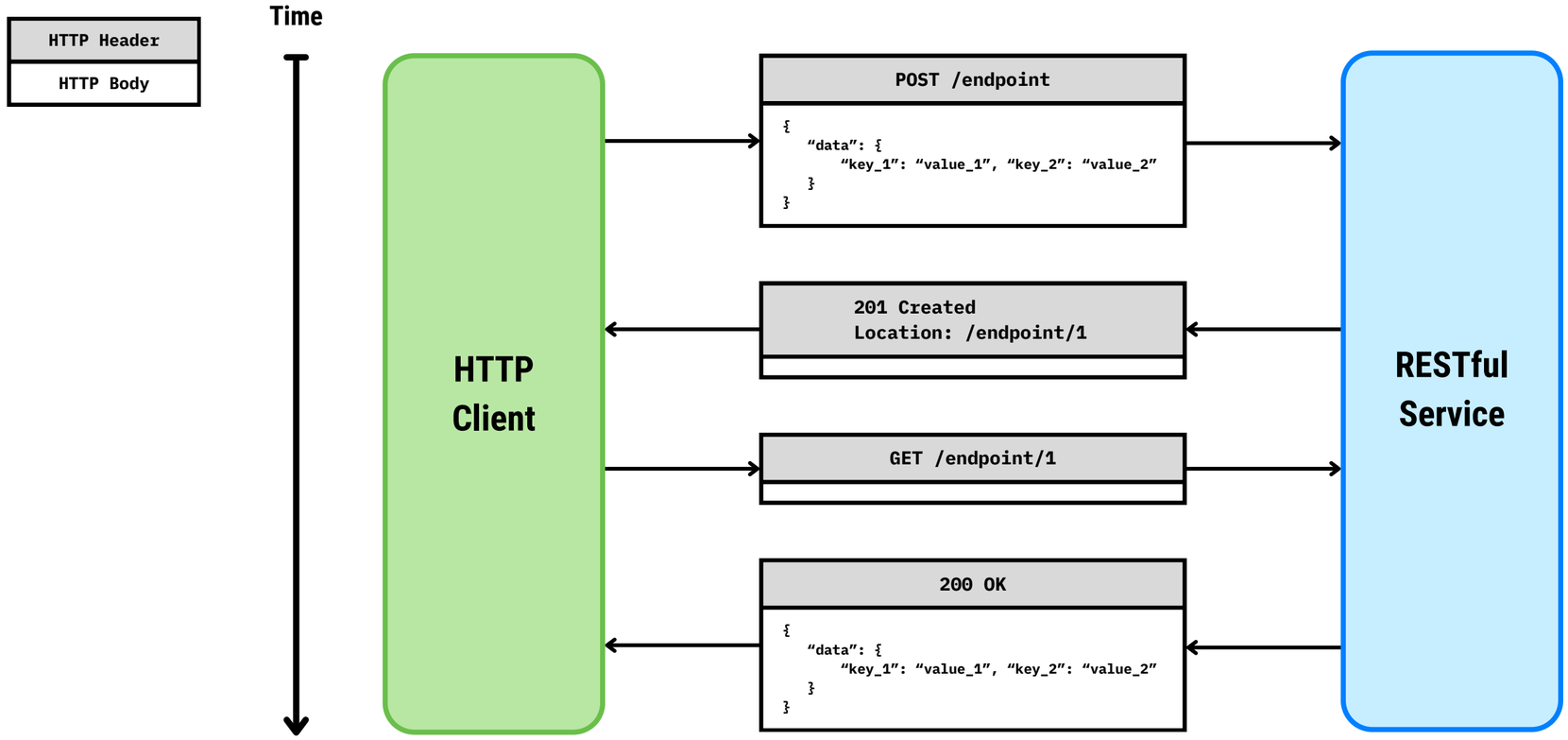
- Protocolo **ligero y binario** para comunicación M2M en la capa de aplicación TCP/IP.
- ***Publish-subscribe:*** clientes publican y se suscriben a tópicos específicos.
- Broker intermediario que enruta los mensajes entre los clientes.



HTTP

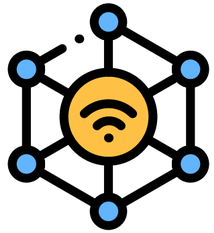


- Protocolo basado en texto para comunicación cliente-servidor en la capa de aplicación TCP/IP.
- ***Request-response:*** clientes envían peticiones al servidor quien responde de vuelta.

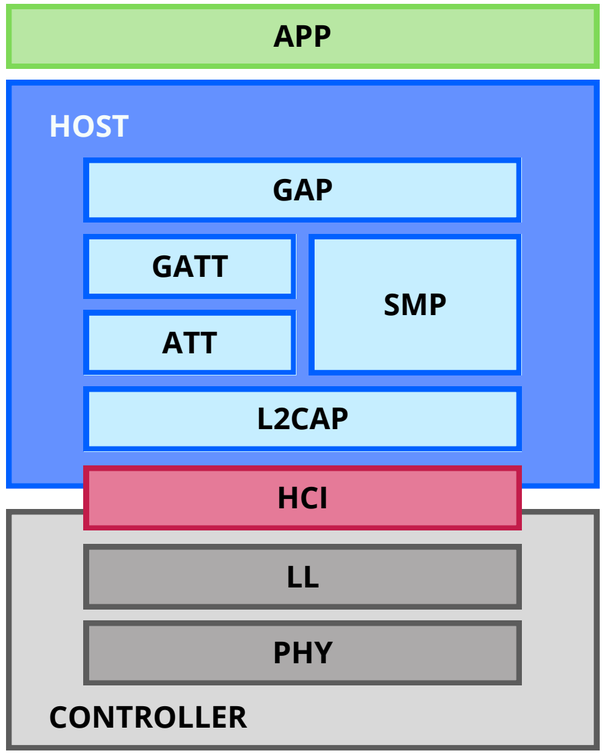


BLE

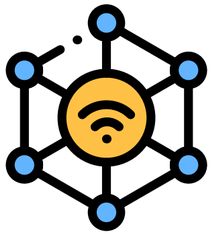
↘



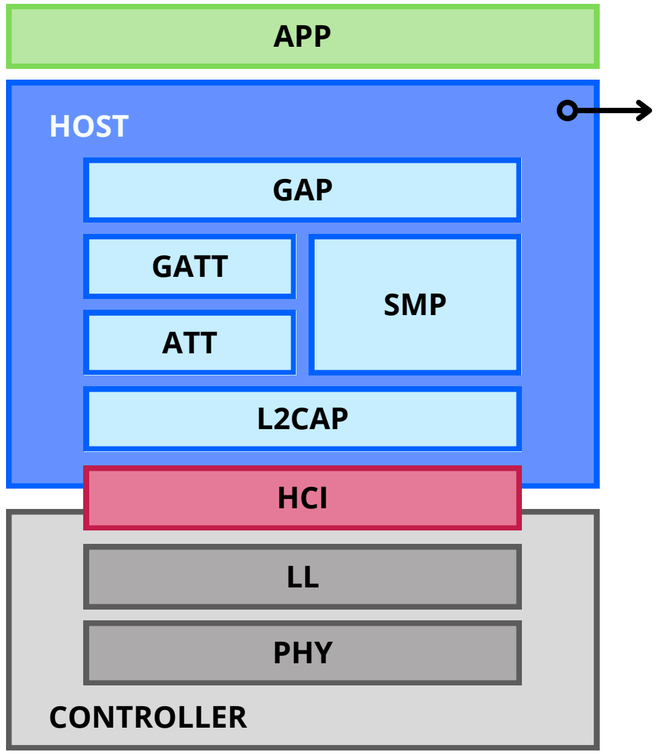
- BLE es una *versión* de Bluetooth de bajo consumo.
- Bluetooth clásico orientado a transmisión continua de datos.
- BLE optimizado para transmisiones breves y esporádicas.



BLE



- BLE es una *versión* de Bluetooth de bajo consumo.
- Bluetooth clásico orientado a transmisión continua de datos.
- BLE optimizado para transmisiones breves y esporádicas.



Host

GAP: Define roles de dispositivos *(Peripheral & Central)*.

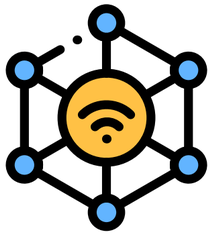
GATT: Provee estructura para acceder a datos *(Services & Characteristics)*.

ATT: Permite el operar sobre *Attributes*.

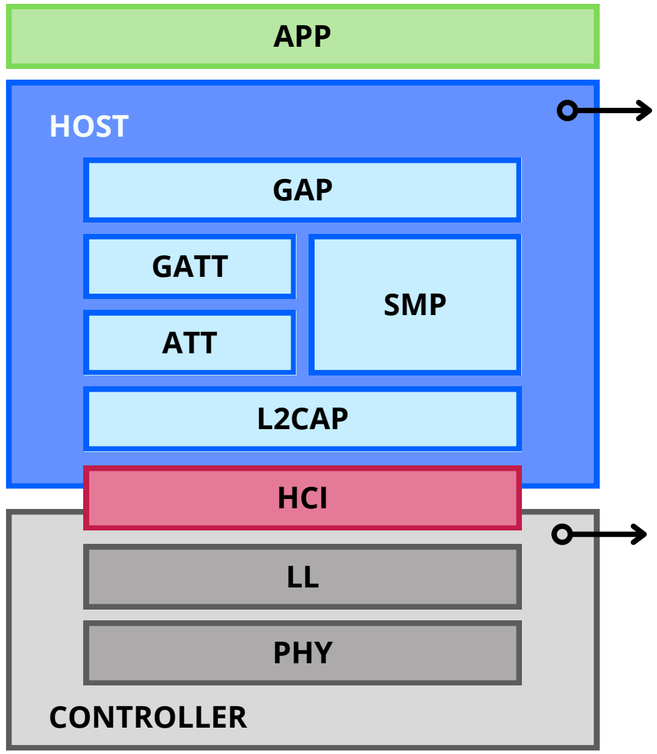
SMP: Gestiona seguridad en BLE, generalmente ignorado!!!

L2CAP: Multiplexion de datos entre capas superiores y capa de enlace.

BLE



- BLE es una *versión* de Bluetooth de bajo consumo.
- Bluetooth clásico orientado a transmisión continua de datos.
- BLE optimizado para transmisiones breves y esporádicas.



Host

GAP: Define roles de dispositivos *(Peripheral & Central)*.

GATT: Provee estructura para acceder a datos *(Services & Characteristics)*.

ATT: Permite el operar sobre *Attributes*.

SMP: Gestiona seguridad en BLE, generalmente ignorado!!!

L2CAP: Multiplexion de datos entre capas superiores y capa de enlace.

Controller

PHY:

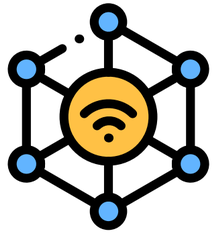
- Opera en la banda ISM de 2.4 GHz.
- TX/RX señales.
- 40 Canales: 37 advertisement y 3 data.

LL:

- Establece y mantiene conexiones.
- Gestiona Advertisement y Scaneo.
- Manejo de paquetes de datos.

GAP

↘



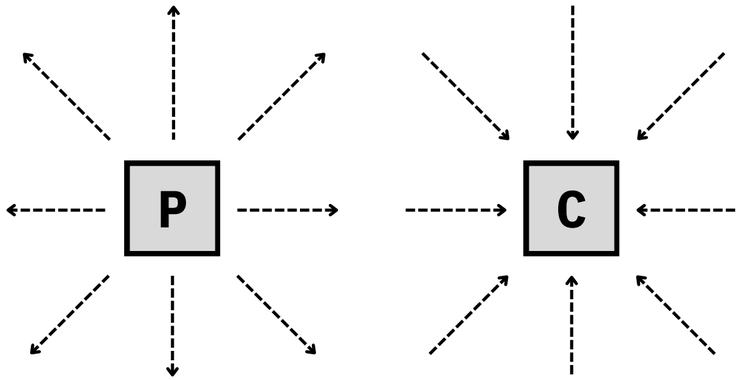
Peripheral

- Avisa de su disponibilidad hasta que un *Central* establece una conexión.

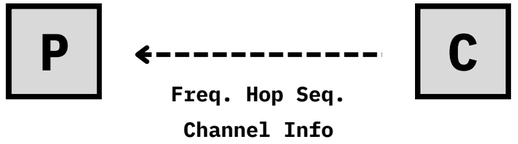
Central

- Escanea buscando *Peripherals* y establece múltiples conexiones.

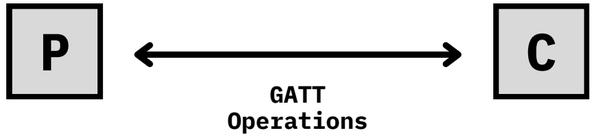
(1) Advertisement - Scanning



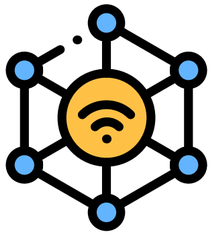
(2) Initiating Connection

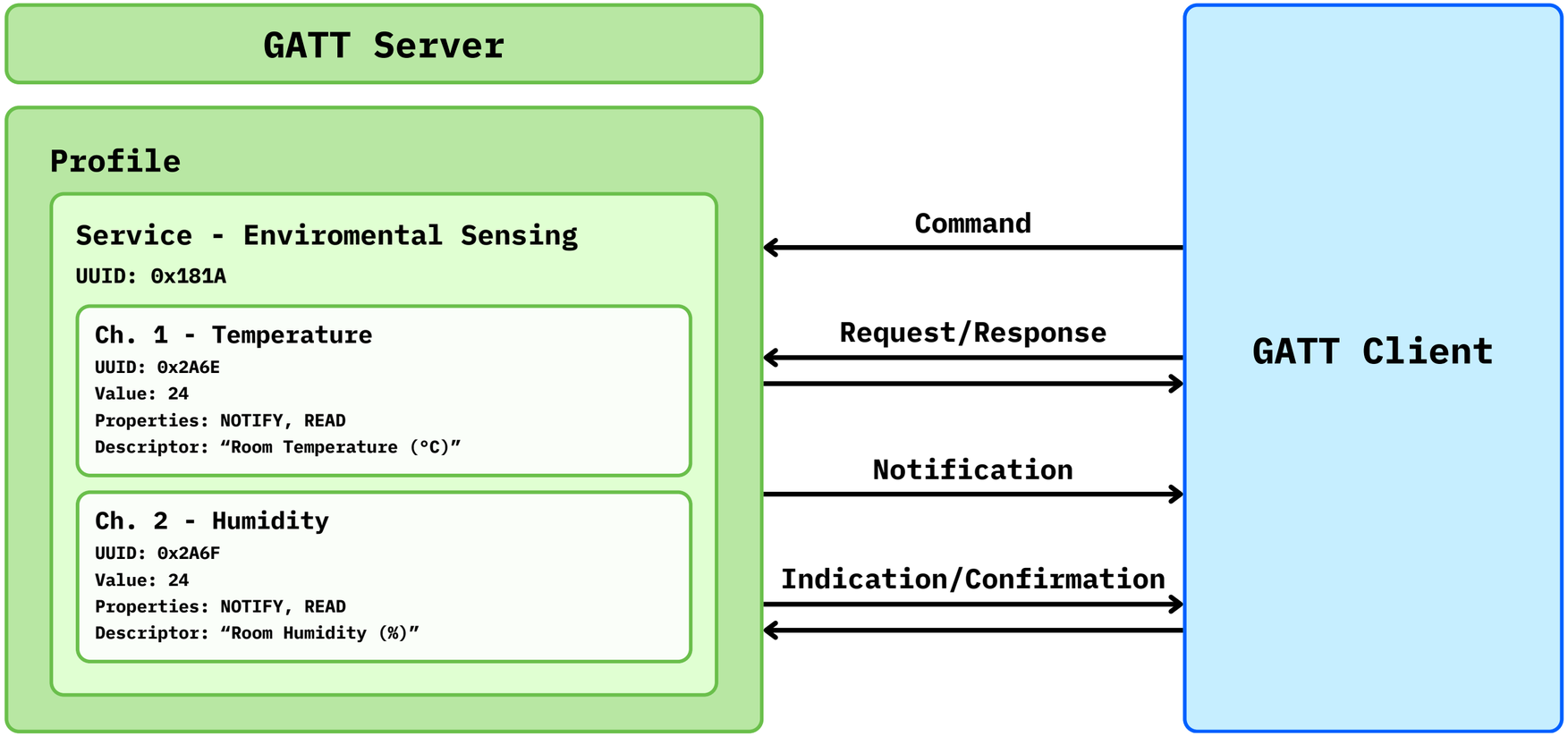


(3) Connection Established



GATT

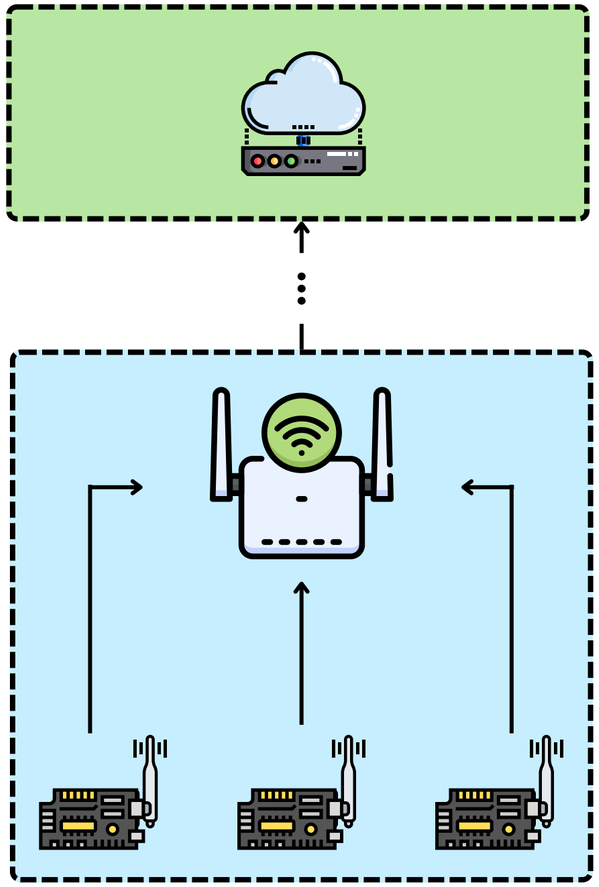



GATT Server
Profile
Service - Enviromental Sensing
UUID: 0x181A
Ch. 1 - Temperature
UUID: 0x2A6E
Value: 24
Properties: NOTIFY, READ
Descriptor: “Room Temperature (°C)”
Ch. 2 - Humidity
UUID: 0x2A6F
Value: 24
Properties: NOTIFY, READ
Descriptor: “Room Humidity (%)”
Command
Request/Response
Notification
Indication/Confirmation
GATT Client

PdM:
Data Processing

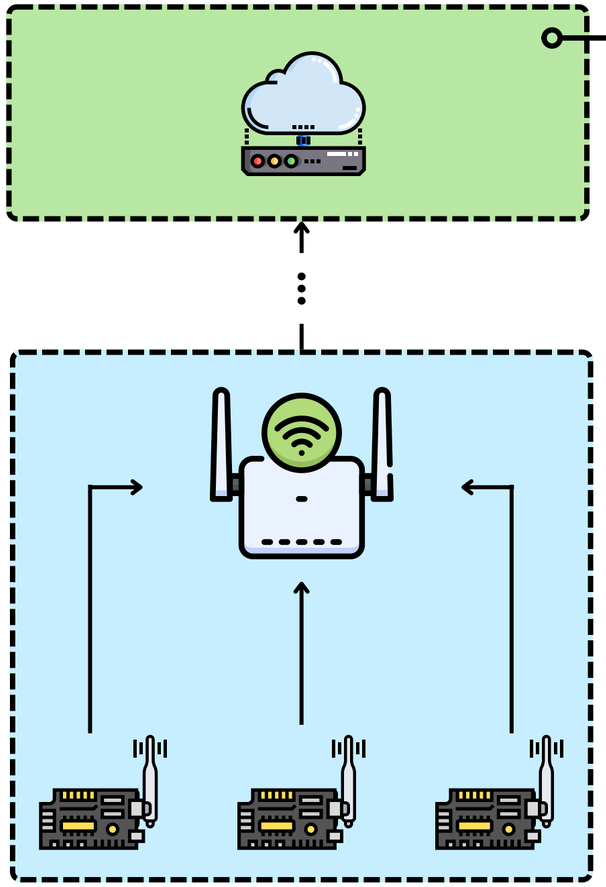
Paradigmas de Computo

Enfoques que definen dónde y cómo se procesan los datos en una red de dispositivos conectados, siendo Cloud Computing y Edge Computing los más importantes [8].



Paradigmas de Computo

Enfoques que definen dónde y cómo se procesan los datos en una red de dispositivos conectados, siendo Cloud Computing y Edge Computing los más importantes [8].

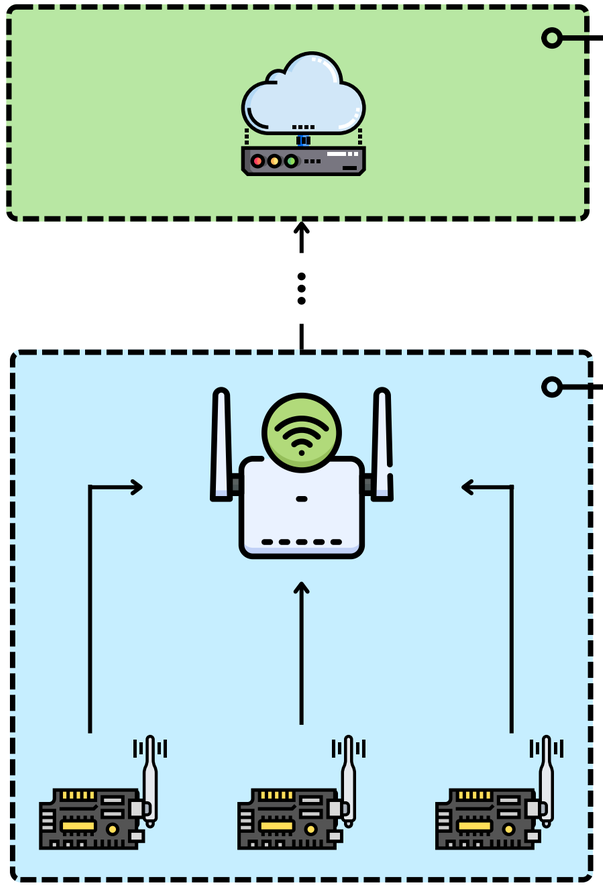


Cloud Computing

- Procesamiento centralizado.
- Problemas de privacidad.
- Costoso al escalar.
- Alto poder de computo.

Paradigmas de Computo

Enfoques que definen dónde y cómo se procesan los datos en una red de dispositivos conectados, siendo Cloud Computing y Edge Computing los más importantes [8].



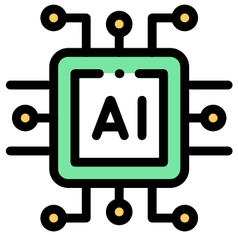
Cloud Computing

- Procesamiento centralizado.
- Problemas de privacidad.
- Costoso al escalar.
- Alto poder de computo.

Edge Computing

- Procesamiento distribuido.
- Cercano a donde se generan los datos.
- Barato al escalar.
- Buena privacidad pero propenso a ataques.
- *Edge devices* restringidos en memoria y CPU.

Inteligencia Artificial

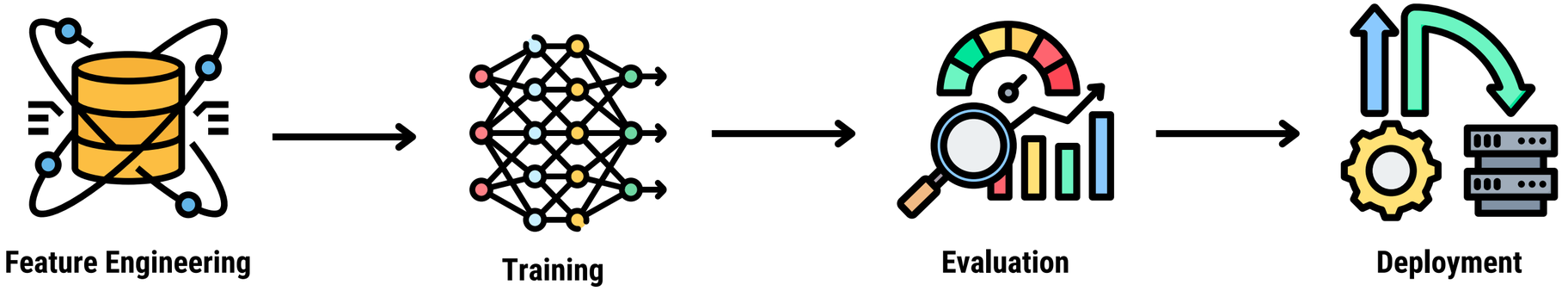


La inteligencia artificial (IA) ha revolucionado el data-driven PdM, permitiendo identificar patrones imposibles de detectar con métodos tradicionales [9].

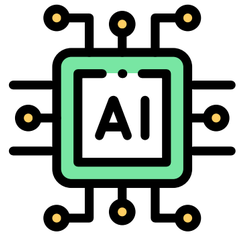
↘

Machine Learning (ML)

- ML permite construir modelos capaces de realizar predicciones sobre datos no vistos.
- Modelos aprenden a partir de **datos estructurados** entregados en el entrenamiento.
- Existen 3 paradigmas del aprendizaje: *Supervised, Unsupervised & Reinforced.*



Inteligencia Artificial



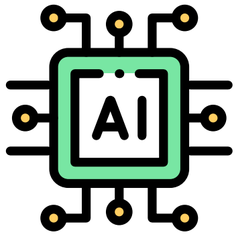
↘

ML en PdM

Aplicación	Supervised	Unsupervised	Reinforced
Detección de Anomalías		✓	
Monitoreo de Condición	✓		
Estimación de la Vida Útil Restante	✓		
Optimización de rutinas de mantenimiento.			✓

Inteligencia Artificial

↘



Deep Learning (DL)

- Modelos de DL consisten en **múltiples capas** de unidades de computo.
- La más sencilla es el **perceptrón**, inspirada por el comportamiento de las neuronas.
- Cada neurona aprende **actualizando sus parámetros** durante el entrenamiento.

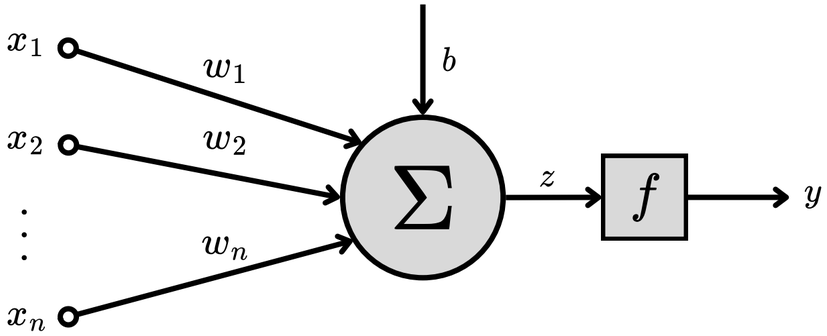
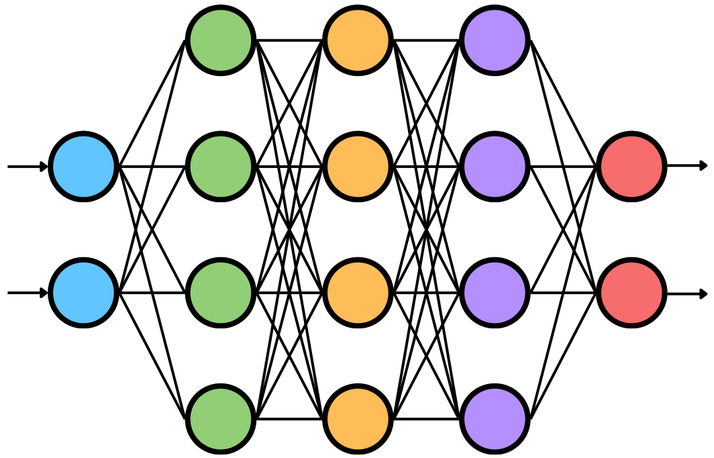
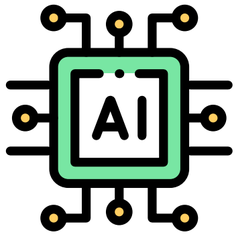


Diagram of a Perceptron



Arquitecturas del DL

↘



- En PdM los datos son generalmente señales digitales.
- *Convolutional Neural Networks (CNN)* permiten capturar **patrones espaciales.**
- *Recurrent Neural Networks (RNN)* permiten capturar **dependencias temporales.**

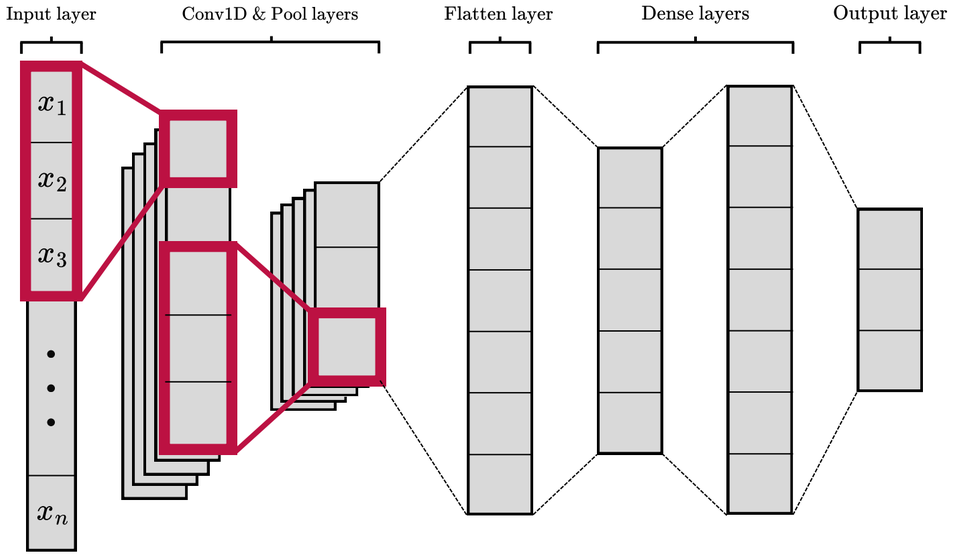


Diagram of a CNN

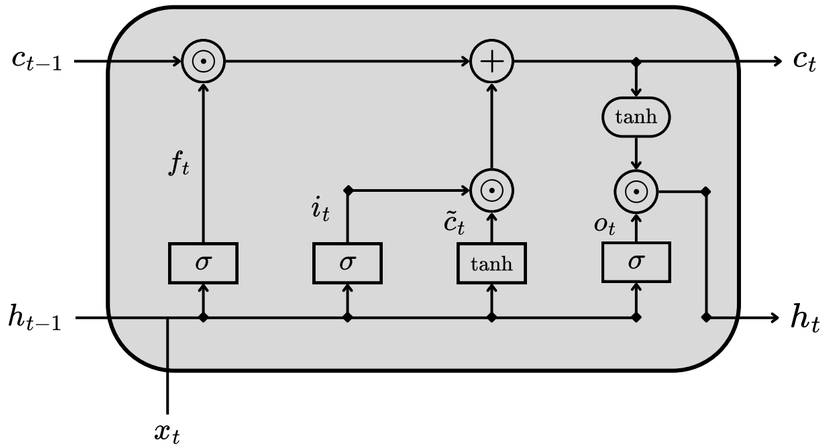


Diagram of an LSTM cell

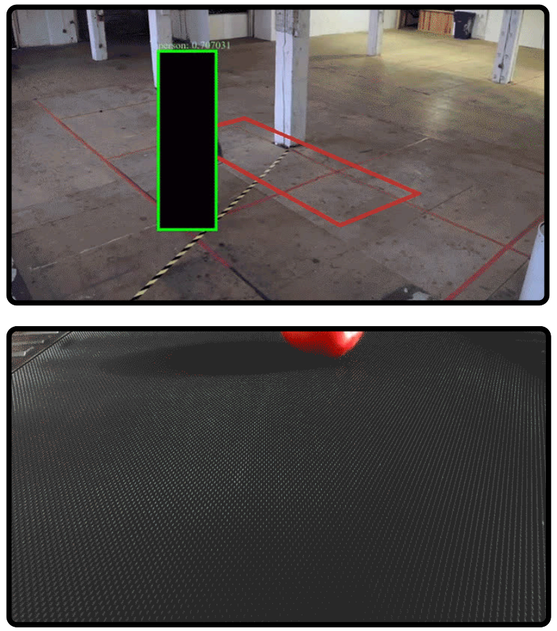
TinyML

Tiny Machine Learning (TinyML) busca habilitar capacidades de ML en dispositivos limitados como wearables, sensores inalámbricos, smartphones, SBCs y MCUs [10].

↘

TinyML existe gracias a constantes innovaciones en hardware y software.

- **Hardware:**
  - *NVIDIA Jetson Nano (SBC).*
  - *Google Coral Edge TPU (ASIC) .*
  - *AMD's Zynq UltraScale+ (FPGA).*
- **Software:**
  - *Tensorflow Lite for Microcontrollers (TFLM).*
  - *Arm's CMSIS NN.*
  - *NVIDIA TensorRT.*



Demo Coral Edge TPU - Google ©

*Nosotros nos centraremos en el software, particularmente a las técnicas de **optimización** de **redes neuronales** en **TFLM**.*

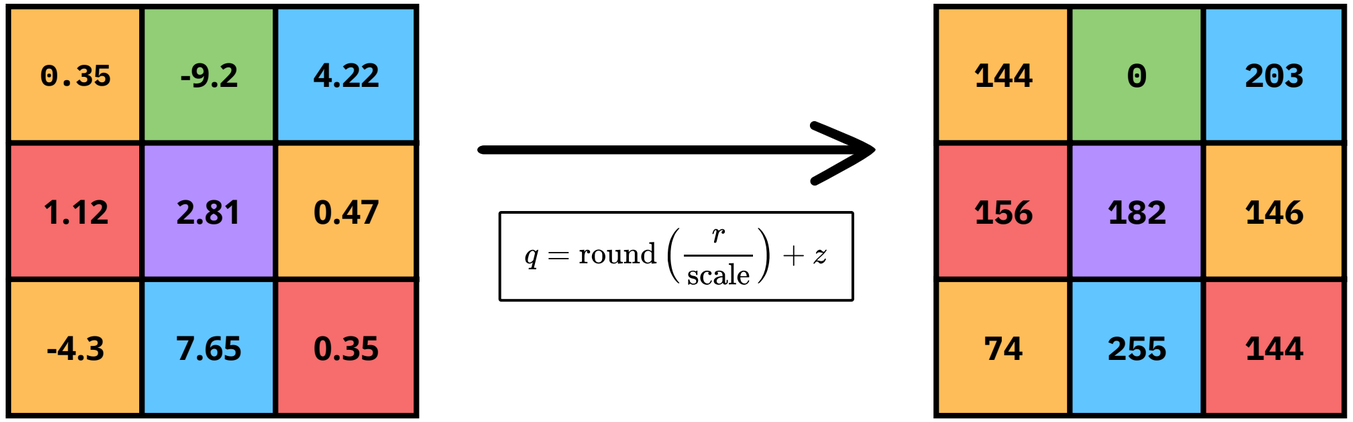
Quantization

Técnica de optimización que busca reducir el tamaño y acelerar el proceso de inferencia de un modelo disminuyendo la precisión de sus componentes de 32 a 8 bits [11].

↘

***Dynamic-range Quantization*:** Cuantización estática para parámetros y dinámica para activaciones, inputs y outputs.

***Full Integer Quantization*:** Cuantización estática de todos los componentes, requiriere conocer la distribución de datos.



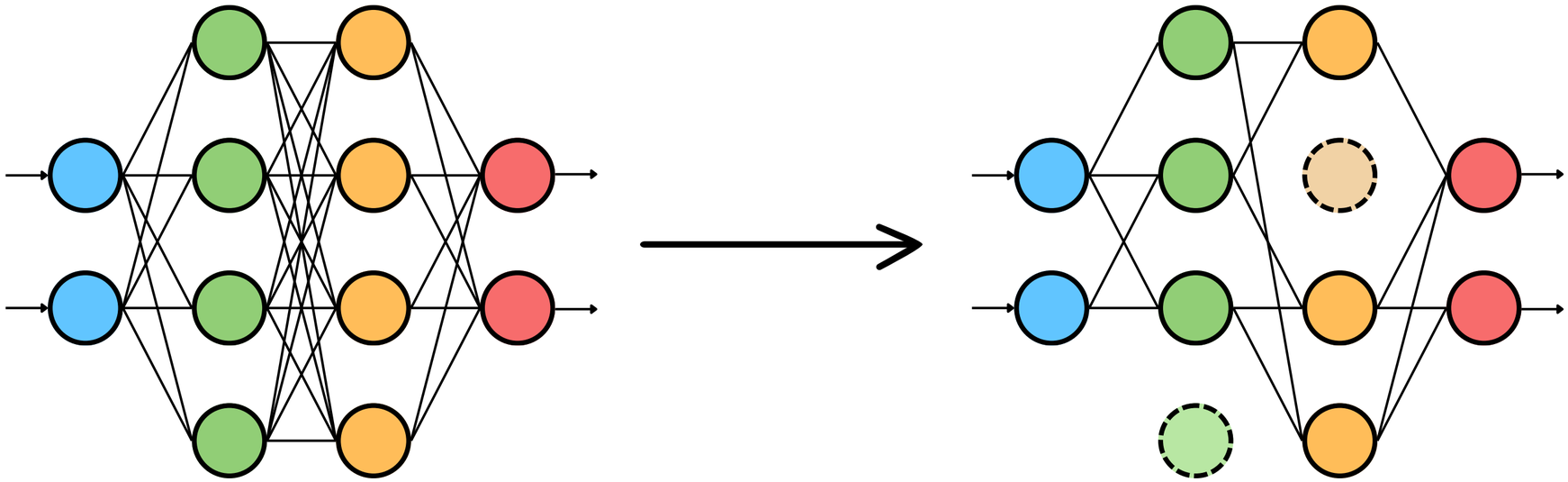
Prunning

Técnica de optimización que progresivamente lleva a cero los parámetros menos relevantes en un modelo, acelerando el proceso de inferencia e indirectamente ayudando a la generalización [12].

↘

Notar que:

- Tras aplicar prunning, el tamaño del modelo es el **mismo**.
- Sin embargo, el tamaño tras compresión es considerablemente menor al caso sin prunning.



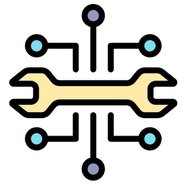
Before Prunning **After Prunning**

PdM Frameworks

PdM Frameworks

Soluciones estructuradas que integran dispositivos físicos, recopilación, visualización y procesamiento de datos, redes, modelos de ML y el personal para permitir el PdM.

↘

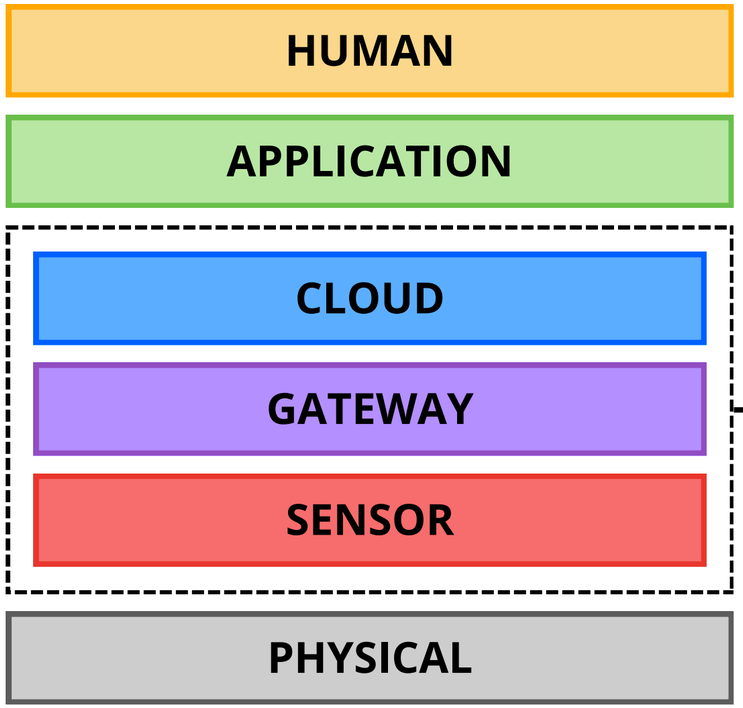


HUMAN	Personal de mantenimiento y operadores de maquinaria.
APPLICATION	Dashboards, aplicaciones móviles y otras UIs.
CLOUD	Infraestructura cloud, bases de datos y APIs.
GATEWAY	Gestión de WSNs.
SENSOR	Firmware de nodos sensor.
PHYSICAL	Diseño electrónico de nodos sensor.

PdM Frameworks

Soluciones estructuradas que integran dispositivos físicos, recopilación, visualización y procesamiento de datos, redes, modelos de ML y el personal para permitir el PdM.

↘



Infraestructura IT

- Núcleo del sistema PdM, encargado del *data acquisition* y *data processing*.
- **¿Donde se despliega el ML?**
  - Cloud-based Inference.
  - Gateway-based Inference.
  - Sensor-based Inference.
- La decisión afecta directamente:
  - *Accuracy* de las predicciones.
  - Latencia de inferencia.
  - Consumo energético nodos.
  - Tráfico en la WSN.
- La decisión se mantiene **fija**.

Problema de Investigación

“La rigidez en el proceso de inferencia exhibido por los PdM frameworks propuestos en la literatura no es suficiente para aquellos escenarios en donde las condiciones operacionales varían en el tiempo, requiriendo una solución más flexible.”

Preguntas de Investigación

↘

Preguntas principales

¿Cómo se puede diseñar un PdM framework que organice modelos de ML de manera jerárquica a través de las tres capas de una red IoT?

- ***Hipótesis 1:*** *Modelos de ML cada vez más complejos pueden desplegarse en un solo PdM framework que integre los acercamientos tradicionales, permitiendo a los nodos elegir dónde realizar la inferencia.*

¿Cómo puede adaptarse la ubicación de la inferencia de un nodo según factores operativos?

- ***Hipótesis 2:*** *Heurísticas desplegadas en cada capa pueden actualizar en el modo de inferencia de un nodo basandose en predicciones anteriores, niveles de batería y disponibilidad del gateway.*

Preguntas de Investigación

↘

Preguntas secundarias

¿Cómo afecta el despliegue de modelos TinyML en nodos al consumo energetico y la vida útil de la batería de estos dispositivos versus delegar la inferencia al gateway o la nube?

- ***Hipótesis 3:*** *Desplegar modelos TinyML en nodos IoT económicos y con recursos limitados reduce significativamente el consumo de energía de los dispositivos en comparación con la ejecución de la inferencia de manera remota.*

¿Cómo se comparan las latencias al realizar la inferencia en nodos, gateways o en el cloud?

- ***Hipótesis 4:*** *Al desplegar modelos PdM en una red IoT, la latencia de inferencia es más baja cuando se realiza en los nodos, aumenta ligeramente al ejecutarse en el gateway y es la más alta al depender de la nube.*

¿Cómo se comporta el tráfico en la WSN al realizar la inferencia en nodos, gateways o en el cloud?

- ***Hipótesis 5:*** *El tráfico de la WSN se minimiza al realizar la inferencia en los nodos, aumenta ligeramente al depender del gateway y drásticamente al realizar la inferencia en la nube.*

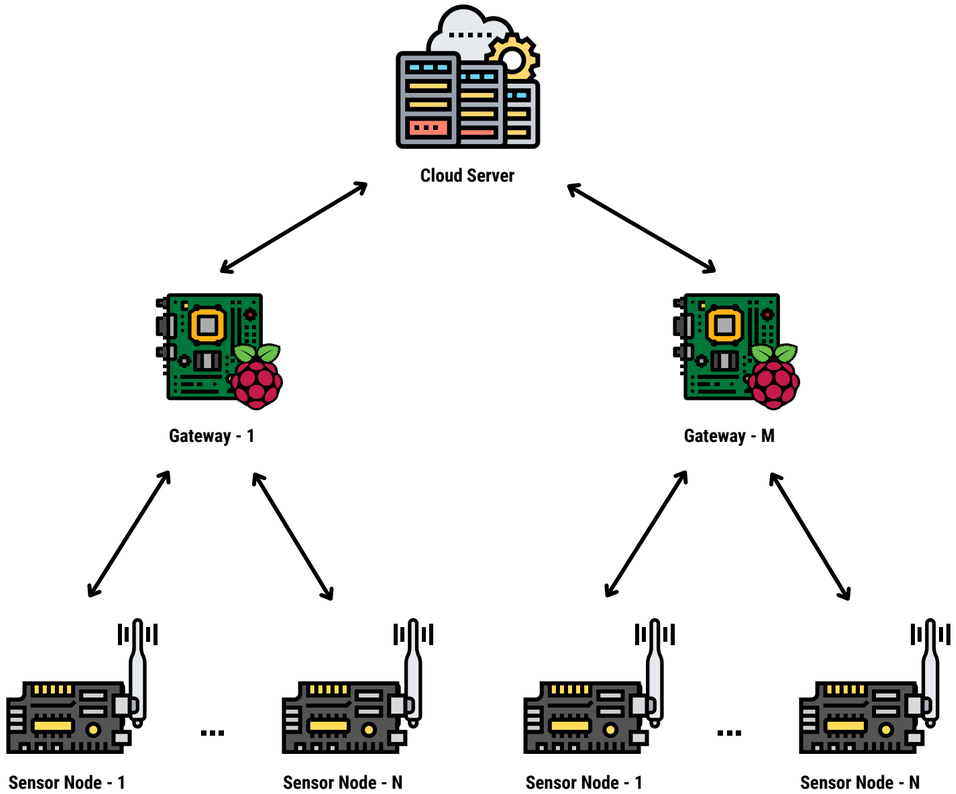
Solucion Propuesta:

ESN-PdM Framework

ESN-PdM Framework

Una solución PdM orientada al monitoreo de condición en tiempo real que combina propuestas tradicionales y las extiende, entregando una infraestructura solida, inferencia jerárquica y versátil.

↘



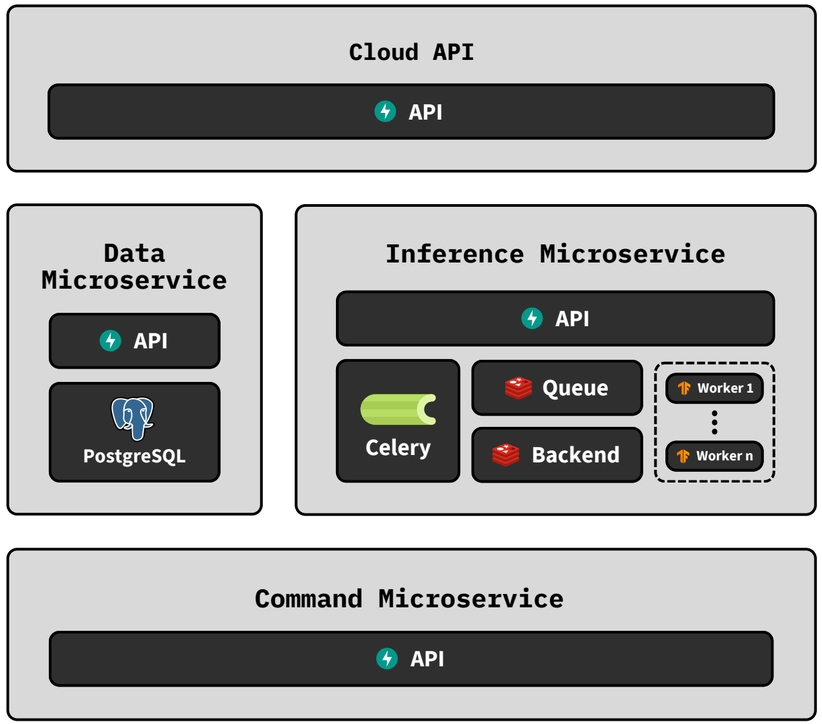
Solución Propuesta

- ***Comprehensive Infrastructure:*** *Ap*rovisionamiento seguro de nodos, la recopilación de datos, la comunicación y el análisis.
- ***Edge and Cloud Intelligence:*** Combina inferencia en sensores, gateway y nube en una solución unificada.
- ***Versatile ML-based PdM:*** Mecanismo adaptativo que actualiza el modo de inferencia en tiempo real.

ESN–PdM: Cloud Layer

Primera capa del framework, interactuando directamente con la capa de aplicacion y la capa gateway.
La capa cloud orquesta el resto del sistema, ofreciendo almacenamiento, inferencia y comandos.

↘



Cloud Layer Architecture Diagram

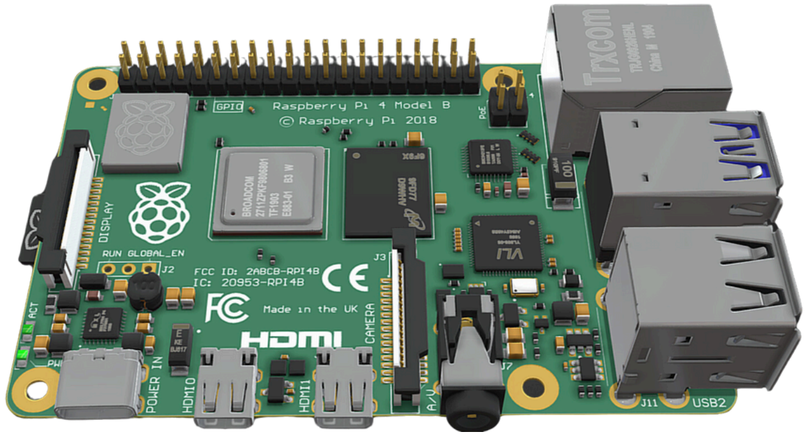
- ***Cloud API:*** Punto de entrada para requests a la capa cloud, proporcionando una API RESTful para interactuar con el sistema.
- ***Data Microservice:*** Maneja y expone operaciones CRUD para los modelos de datos de la capa, interactuando con una base de datos.
- ***Inference Microservice:*** Gestiona una cola de tareas asíncronas. Cada tarea es una medición de un sensor a la espera de inferencia.
- ***Command Microservice:*** Envía comandos a capas inferiores. Almacena respuestas en un caché para rápida recolección.

ESN-PdM: Gateway Layer

Segunda capa del framework, interactúa directamente con la capa cloud y la capa sensor. Gestiona una WSN descubriendo, aprovisionando y configurando nuevos dispositivos, entre otras funciones.

↘

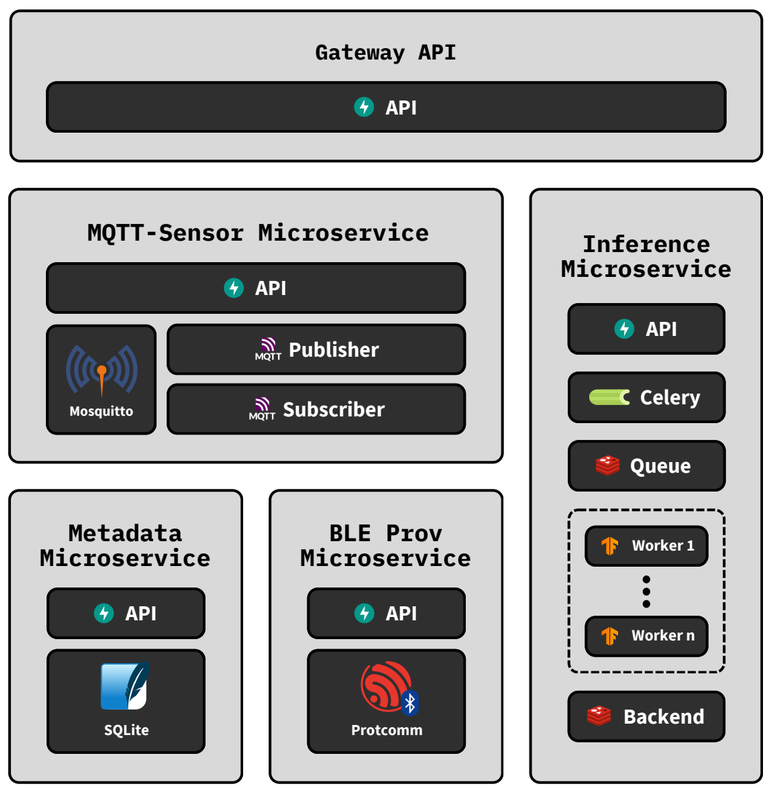
- **Conectividad avanzada:** Incluye Ethernet Gigabit, Wi-Fi 5 y Bluetooth 5.0.
- **Alto rendimiento:** Equipado con un procesador Quad-core Cortex-A72 a 1.5 GHz y hasta 8 GB de RAM.
- **Costo accesible:** SBC muy económico, ofreciendo una excelente relación calidad-precio.
- **Flexibilidad de expansión:** Dispone de un conector GPIO de 40 pines y múltiples puertos USB.
- **Almacenamiento expandible:** Soporta tarjetas microSD de alta capacidad.



Raspberry Pi 4 Model B 3D Render

ESN-PdM: Gateway Layer

Segunda capa del framework, interactúa directamente con la capa cloud y la capa sensor. Gestiona una WSN descubriendo, aprovisionando y configurando nuevos dispositivos, entre otras funciones.



Gateway Layer Architecture Diagram

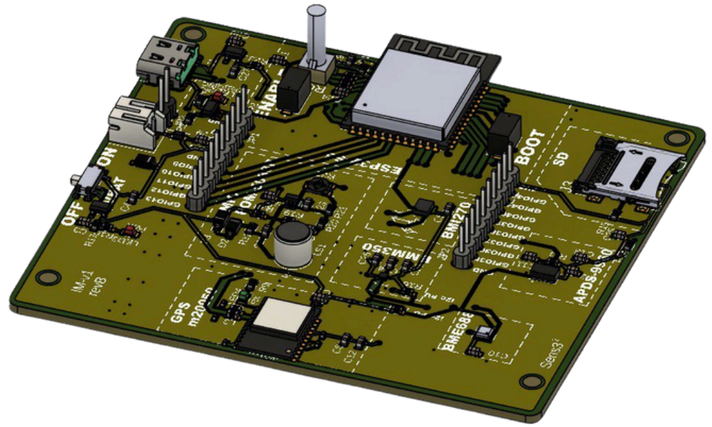
- ***Gateway API:*** Punto de entrada para requests a la capa gateway, proporcionando una API RESTful para interactuar con los servicios.
- ***Metadata Microservice:*** Gestiona todas las operaciones CRUD relacionadas con los metadatos dentro de la capa de gateway.
- ***Inference Microservice:*** Gestiona una cola de tareas asíncronas. Cada tarea es una medición de un sensor a la espera de inferencia.
- ***MQTT-Sensor Microservice:*** Encargado de la comunicación entre el gateway y los sensores a través de MQTT.
- ***BLE Prov Microservice:*** Encargado de detectar y aprovisionar nuevos nodos con las credenciales WiFi a través de BLE utilizando el *Protcomm Provisioning Protocol* de *Espressif*.

ESN–PdM: Sensor Layer

Tercera y ultima capa del framework, envía información a la capa gateway e interactúa directamente con el entorno físico a través de módulos de sensado.

↘

- **MCU:** Microcontrolador ESP32, cuenta con un procesador dual-core a 240 MHz, comunicación Wi-Fi y Bluetooth.
- **Modulo de Sensado:** Bosch BMI270, una IMU capaz de capturar aceleraciones y velocidades angulares en tiempo real.
- **Almacenamiento:** SRAM de 520 KB, flash interna de 16 MB y puerto microSD para flash externa.
- **Alimentación:** Batería Lithium-Polymer (LiPo) de 1400mAh

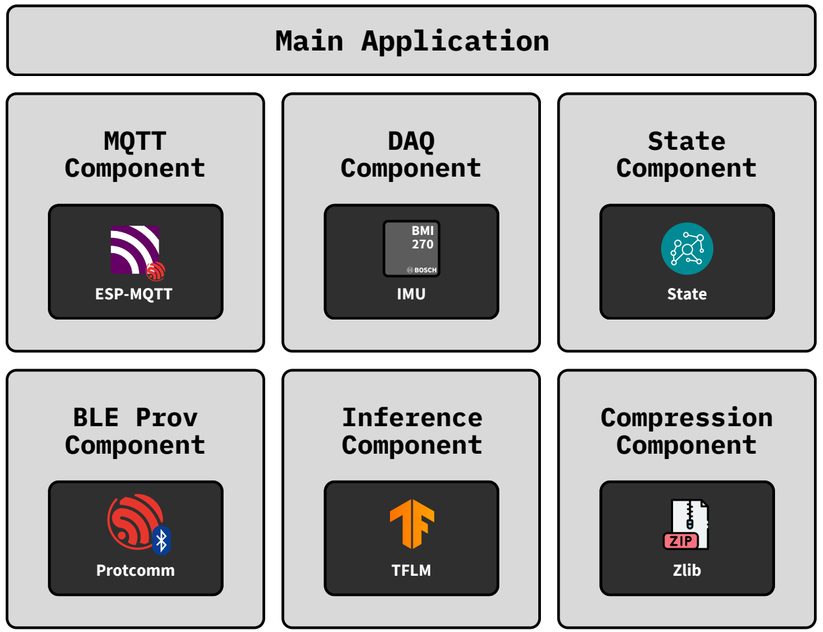


Sensor Node 3D Render

ESN-PdM: Sensor Layer

Tercera y ultima capa del framework, envía información a la capa gateway e interactúa directamente con el entorno físico a través de módulos de sensado.

↘



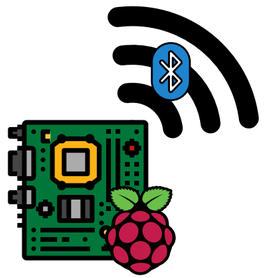
Sensor Layer Firmware Diagram

- ***MQTT Component:*** Maneja la comunicación con la capa de gateway a través de MQTT, subscribiéndose y publicando en topicos.
- ***DAQ Component:*** Encargado de recopilar datos de vibración utilizando el IMU BMI270, comunicándose a través de I²C.
- ***State Component:*** Abstracción del nodo, gestionando la máquina de estados, configuraciones, ejecución de comandos y el ciclo.
- ***BLE Prov Component:*** Implementa *Protcomm* en el nodo, manejando la recepción y verificación de credenciales.
- ***Inference Component:*** Utiliza TFLM para ejecutar un modelo TinyML, realizando inferencias locales.
- ***Compression Component:*** Encargado de comprimir datos crudos de los sensores antes de su transmisión al gateway usando Zlib.

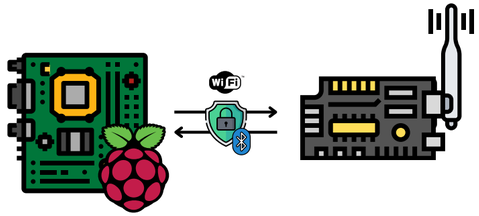
ESN–PdM: Workflow

↘

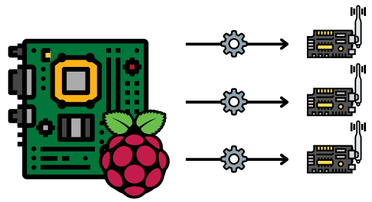
Una solución PdM orientada al monitoreo de condición en tiempo real que combina propuestas tradicionales y las extiende, entregando una infraestructura solida, inferencia jerárquica y versátil.

1. 

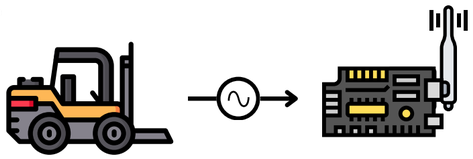
Node Discovery

2. 

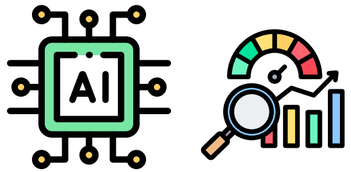
Device Provisioning

3. 

WSN Configuration

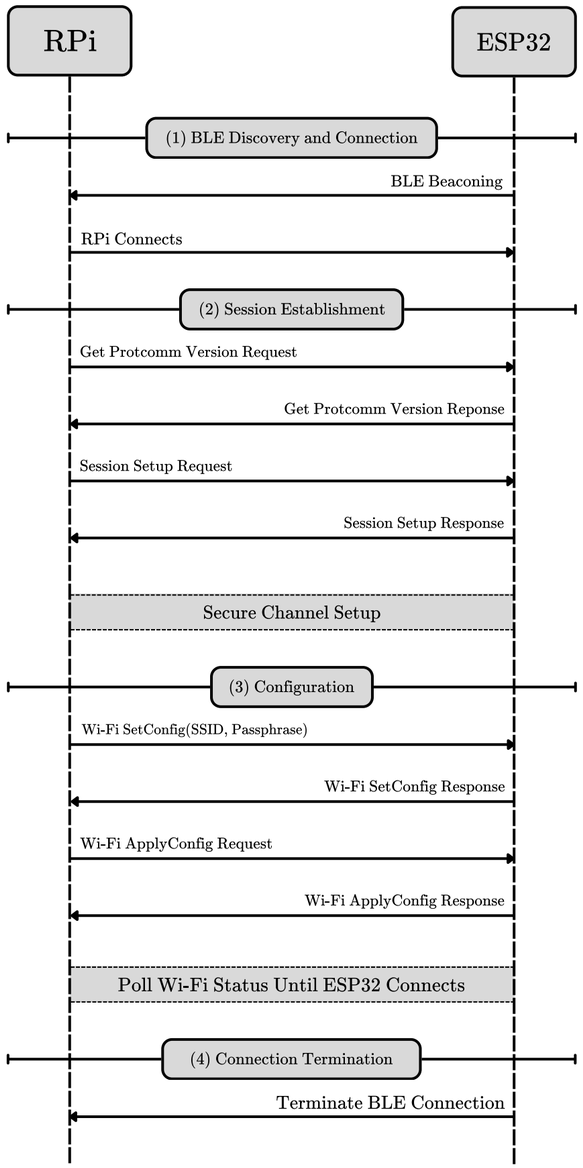
4. 

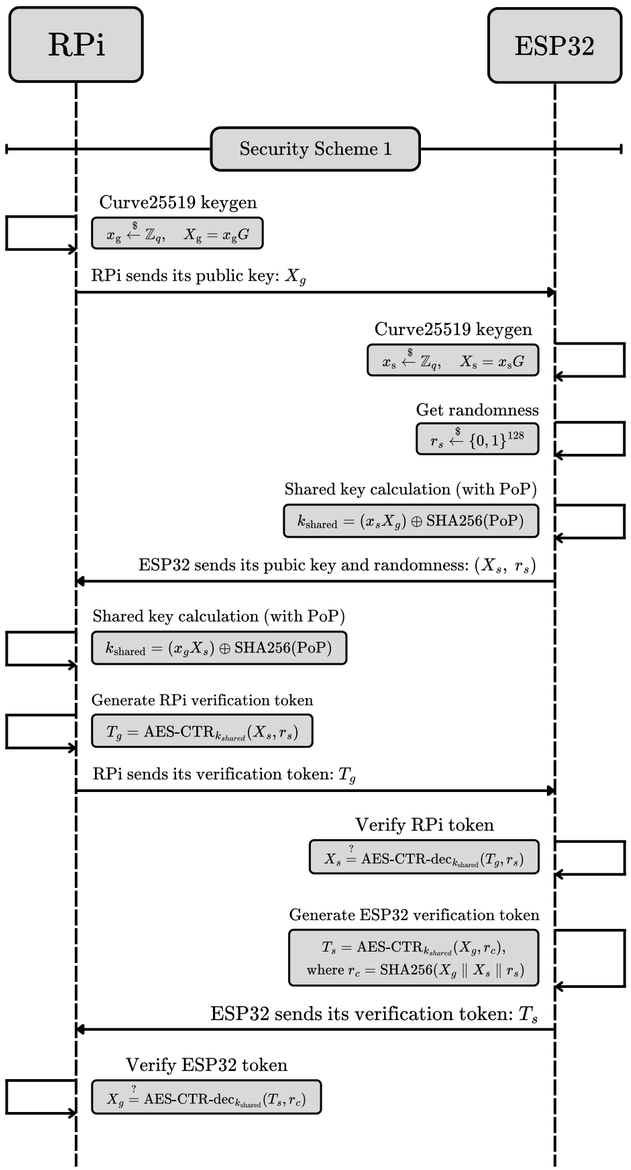
Data Collection

5. 

Inference & Analysis

Descubrimiento de Nodos y Aprovisionamiento





Configuración de la WSN

↘

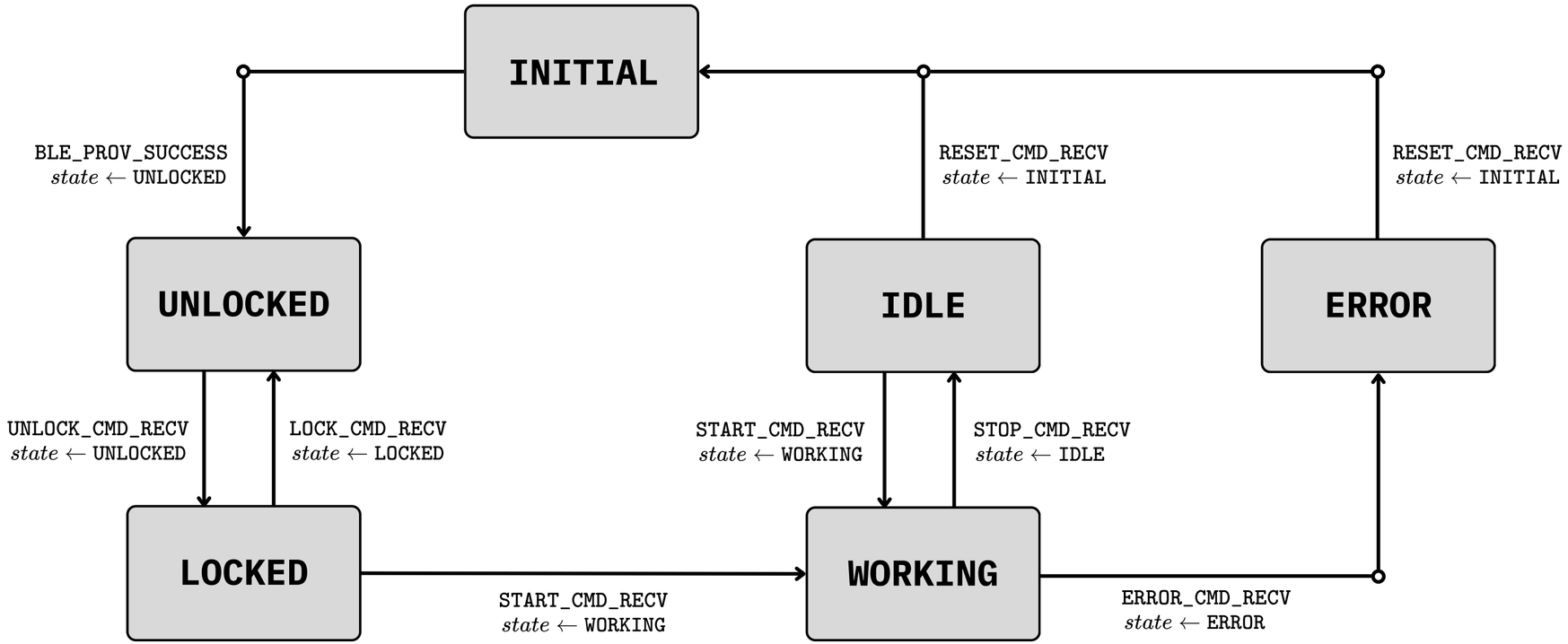
Propiedades operables

Propiedad	Contenido	Métodos permitidos	Dispositivo
tf_model_bytes	Modelo de TF Lite comprimido en gzip y codificado en base64	SET	Gateway/Sensor
tf_model_size	Tamaño original del modelo de TF Lite	SET	Gateway/Sensor
provisioned_nodes	Listado de nodos aprovisionados	GET/ADD	Gateway
inference_mode	Donde un nodo envía datos para inferencia	SET/GET	Sensor
node_state	Estado actual de un nodo	SET/GET	Sensor

Configuración de la WSN

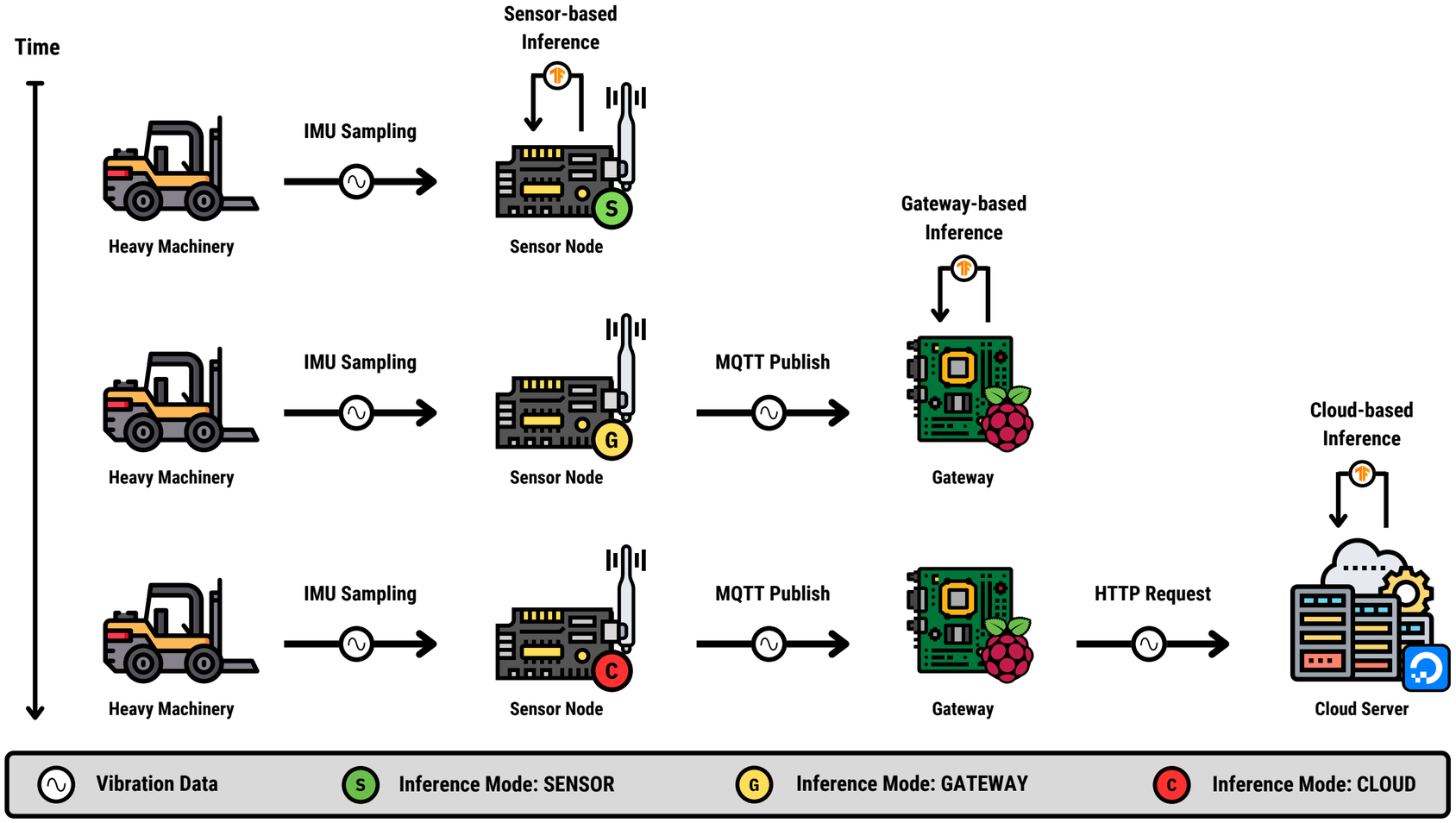
↘

Maquina de Estados del Nodo



Recolección de Datos e Inferencia

Cada nodo cuenta con un **modo de inferencia** el cual puede ser *Sensor*, *Gateway* o *Cloud*. Este modo define que capa del sistema se encargara de emitir una predicción a partir de sus datos.

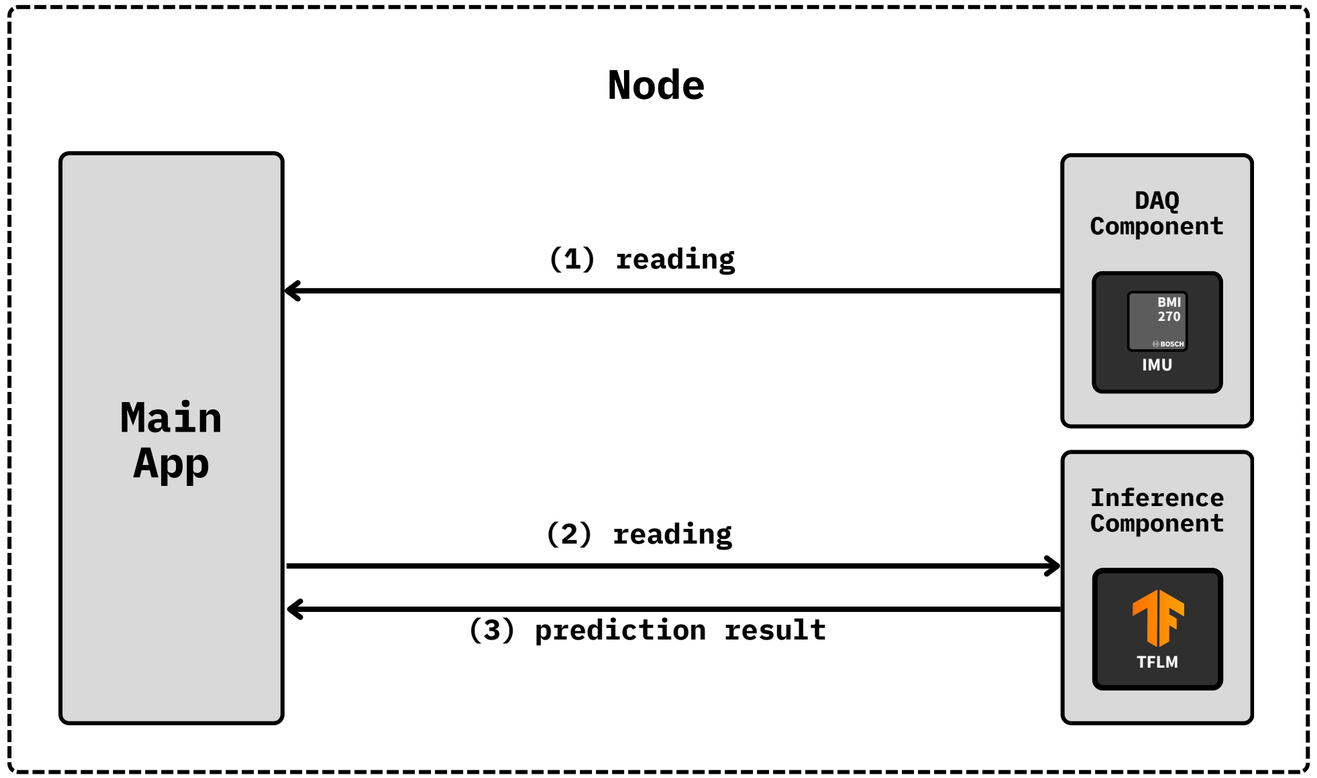


Recolección de Datos e Inferencia

↘

Inferencia en el Sensor

La *main app* solicita una medición al *DAQ Component*. Esta medición se realiza en unidades reales, ya que el modelo en el *Inference Component* así las requiere.

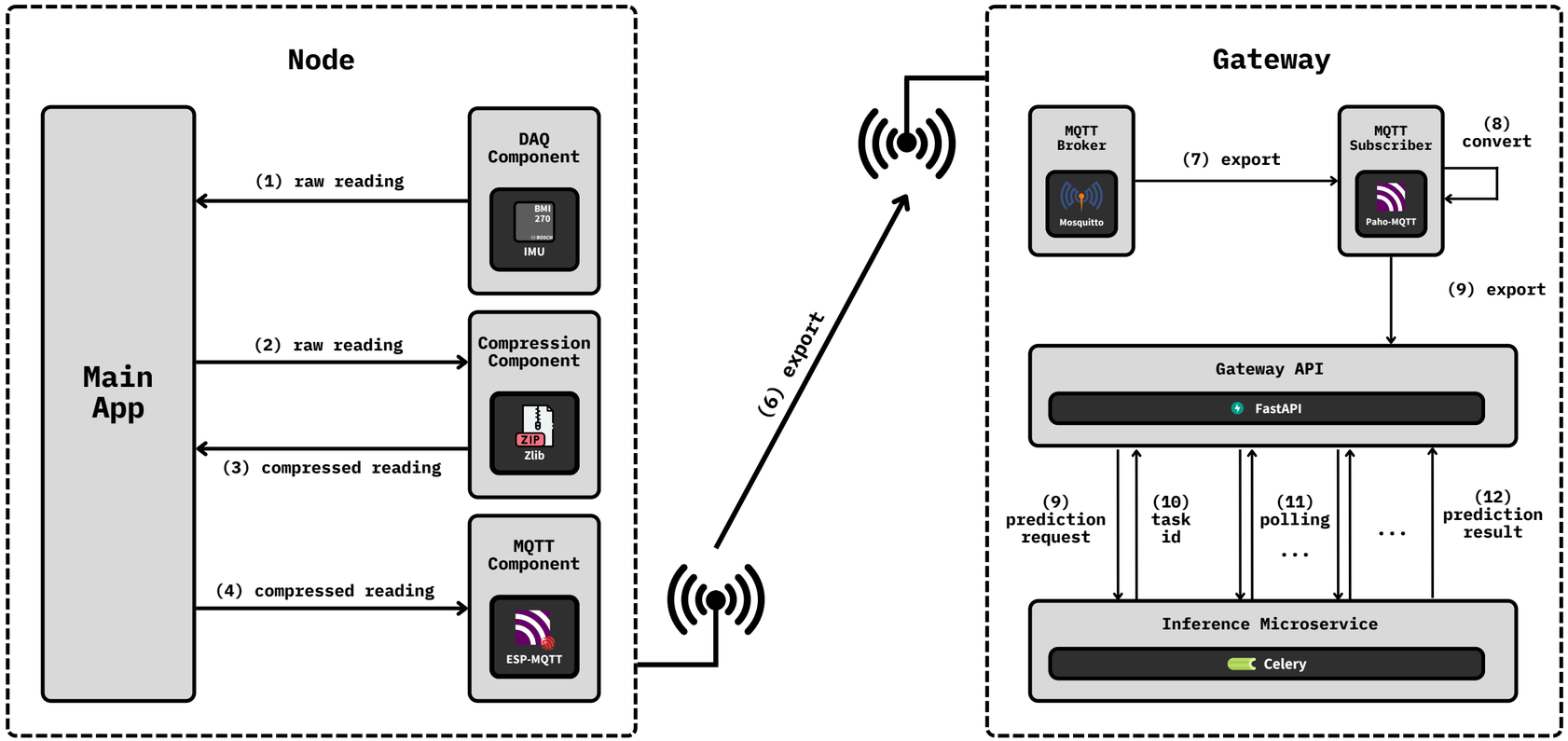


Recolección de Datos e Inferencia

↘

Inferencia en el Gateway

La *main app* solicita una medición cruda al DAQ Component, que se comprime y codifica en base64 antes de enviarse al gateway. Allí se convierte y dirige hasta el *Inference Microservice*.

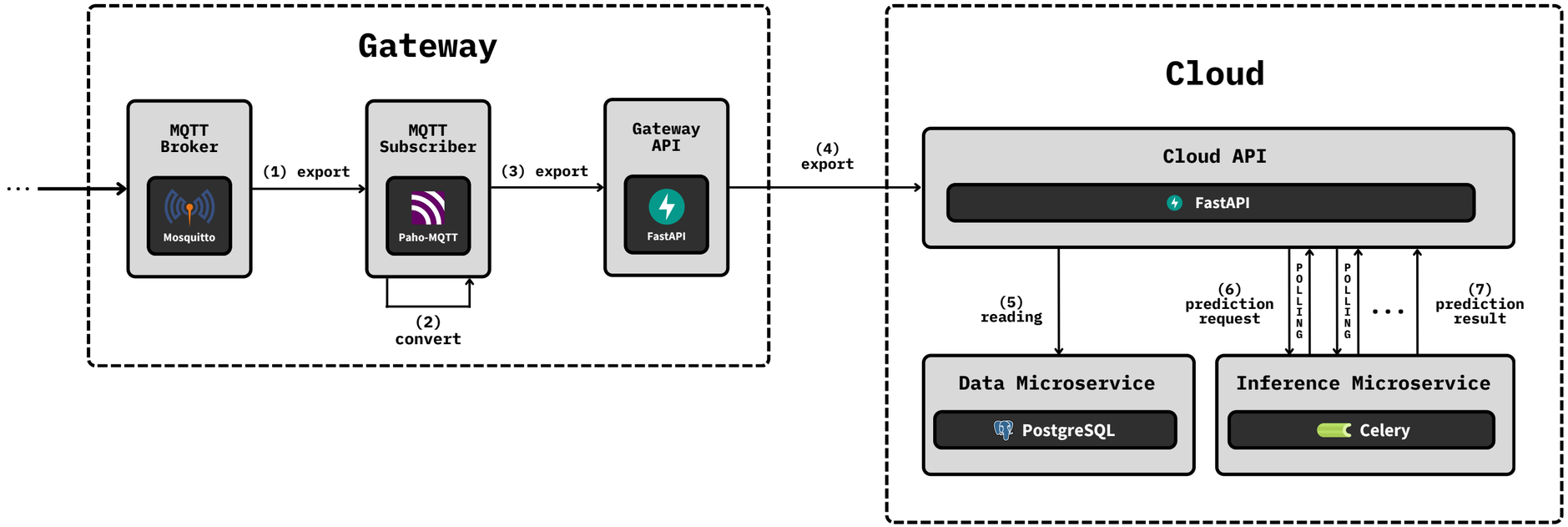


Recolección de Datos e Inferencia

↘

Inferencia en el Cloud

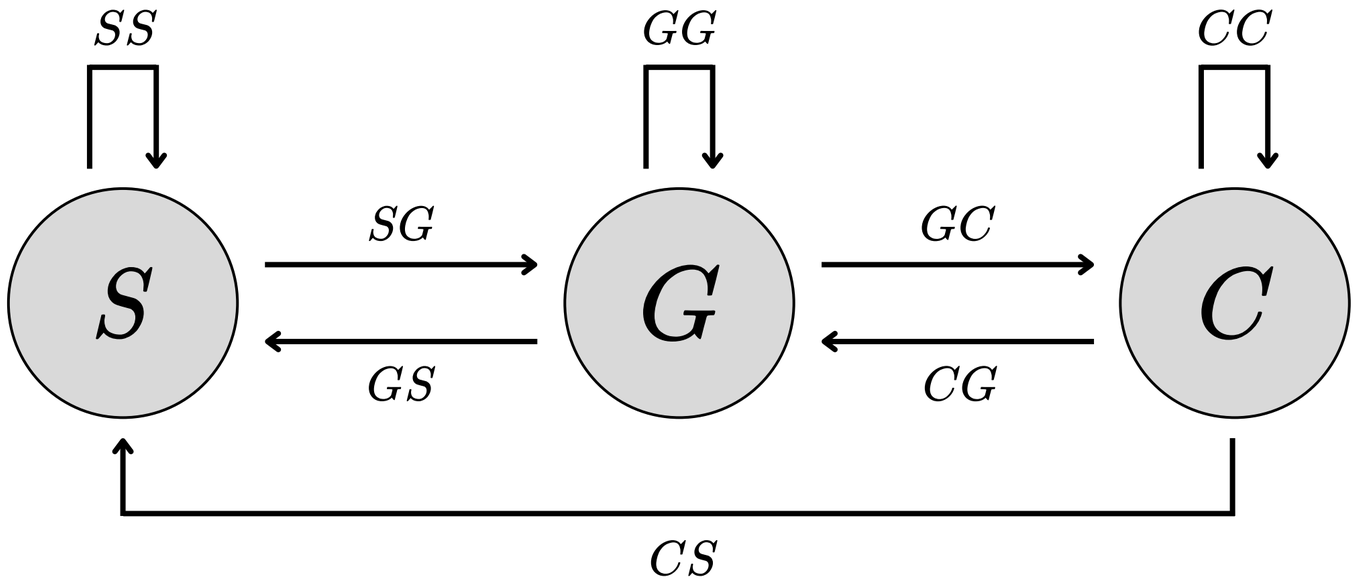
El nodo envía la medición al gateway quien la redirige al Cloud API. Una vez receptado, el API envía al *Data Microservice* la medición y solicita una predicción al *Inferece Microserviece*.



Mecanismo Adaptativo

↘

- Los nodos envían datos a una capa de la red según el modo de inferencia.
- La capa realiza la predicción y evalúa si debe cambiar el modo de inferencia del nodo.
- La decisión de transición la toma una heurística específica de la capa.



Mecanismo Adaptativo

↘

Historial de Anomalias

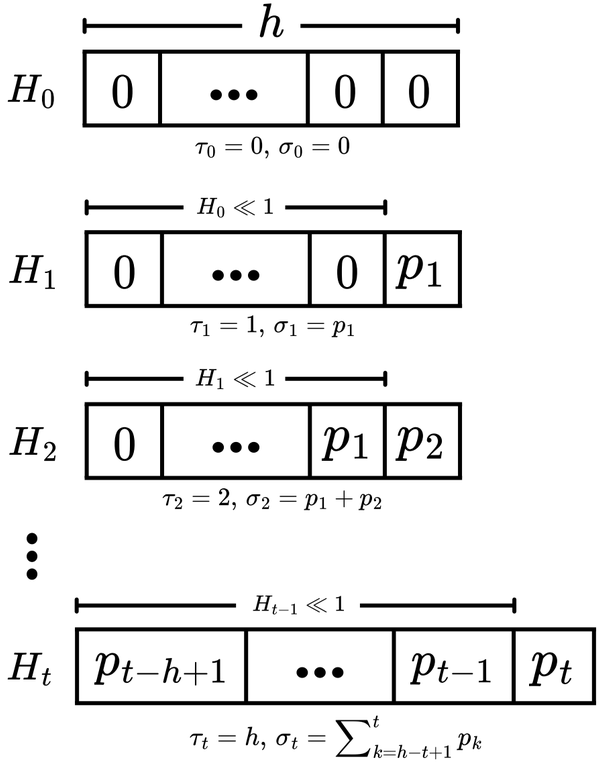
Estructura de datos que almacena las últimas predicciones emitidas por una capa para un nodo en especifico.

$$H_t = \begin{cases} (H_{t-1} \ll 1)\ \&\ p_t & \text{if } X_t = X_{t-1} \\ 0 & \text{otherwise} \end{cases}$$

$$\tau_t = \begin{cases} min(h, \tau_{t-1} + 1) & \text{if } X_t = X_{t-1} \\ 0 & \text{otherwise} \end{cases}$$

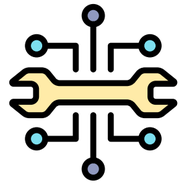
$$\sigma_t = \sum_{i=0}^{h-1} ((H_t \gg i)\ \&\ 1)$$

p_t : Prediccion instante t.
h : Capacidad maxima historial.
τ_t : Largo historial instante t.
σ_t : Anomalias historial instante t.
X_t : Modo inferencia instante t.
H_t : Historial de anomalias instante t.



Mecanismo Adaptativo

↘



Heurística del Sensor

Controla transiciones desde el modo Sensor hacia el modo Gateway.

Algorithm 1 Sensor Adaptive Heuristic

Require: $(X_{t-1} = X_t = S) \wedge p_t, H_{t-1}, b_t, \psi_b, \psi_s, h$
Ensure: The next inference mode X_{t+1}

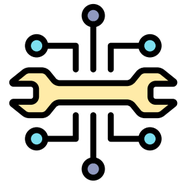
$H_t \leftarrow (H_{t-1} \ll 1)\,\&\,p_t$
$\tau_t \leftarrow \min(h, \tau_{t-1} + 1)$
$\sigma_t \leftarrow \sum_{i=0}^{h-1} ((H_t \gg i)\ \&\,1)$
if $b_t < \psi_b$ **then**
    return S
else if $\tau_t < h$ **then**
    return S
else if $\sigma_t \geq \psi_s$ **then**
    return G
else
    return S
end if

ψ_b : Limite % bateria restante
ψ_s : Limite superior anomalias.

- **Bateria baja:** imposibilidad de escalar modo.
- **Modelo del sensor menos confiable:** genera falsos positivos.
- **Historial amplio y límite bajo:** facilita el escalado al gateway.

Mecanismo Adaptativo

↘



Heurística del Gateway

Se encarga de las transiciones desde el modo Gateway hacia el modo Sensor o modo Cloud.

Algorithm 2 Gateway Adaptive Heuristic

Require: $(X_{t-1} = X_t = G) \wedge p_t, H_{t-1}, q_t, \psi_g, \phi_g, \psi_q, h$
Ensure: The next inference mode X_{t+1}

$H_t \leftarrow (H_{t-1} \ll 1)\,\&\,p_t$
$\tau_t \leftarrow \min(h, \tau_{t-1} + 1)$
$\sigma_t \leftarrow \sum_{i=0}^{h-1} ((H_t \gg i)\ \&\ 1)$
if $b_t < \psi_b$ **then**
    return S
else if $\tau_t < h$ **then**
    return G
else if $\sigma_t < \phi_g$ **then**
    return S
else if $\phi_g \leq \sigma_t < \psi_g \wedge q_t < \psi_q$ **then**
    return G
else
    return C
end if

ψ_b : Limite % bateria restante
ϕ_g : Limite inferior anomalias.
ψ_g : Limite superior anomalias.
q_t : Largo cola prediccion Gateway instante t.
ψ_q : Limite largo cola prediccion.

- **Bateria baja:** de-escalar modo para ahorrar batería.
- **Disponibilidad del gateway:** Si la cola esta llena, se delega inferencia al cloud.
- **Historial más pequeño y límite superior alto:** solo escala a Cloud si hay muchas anomalias seguidas.

Mecanismo Adaptativo

↘

Heurística del Cloud

Gestiona transiciones desde el modo Cloud hacia el modo Gateway.

Algorithm 3 Cloud Adaptive Heuristic

Require: $(X_{t-1} = X_t = C) \wedge p_t, H_{t-1}, h$
Ensure: The next inference mode X_{t+1}

$H_t \leftarrow (H_{t-1} \ll 1)\,\&\,p_t$
$\tau_t \leftarrow \min(h, \tau_{t-1} + 1)$
$\sigma_t \leftarrow \sum_{i=0}^{h-1} ((H_t \gg i)\ \&\,1)$
if $b_t < \psi_b$ **then**
    return S
else if $\tau_t < h$ **then**
    return C
else if $\sigma_t < \phi_c$ **then**
    return G
else
    return C
end if

ψ_b : Limite % bateria restante
ψ_s : Limite superior anomalias.

- **Bateria baja:** de-escalar modo para ahorrar batería.
- **Historial aún más pequeño y límite inferior bajo:** Solo la actividad anómala real llega a la nube y se desescala cuando no hay anomalías.

Caso de Estudio:

Monitoreo de Condición Operacional de Grúas Horquilla en la Minería usando DL.

Caso de Estudio

Monitoreo de Condición Operacional de Grúas Horquilla en la Minería usando DL.

↘

Importancia

- Soporte logístico de las operaciones.
- Transporte de equipos, menas, suministros esenciales, etc.

Desafíos Operativos

- Circunstancias ambientales y climáticas hostiles.
- Conectividad intermitente en yacimientos aislados.
- Experiencia operador afecta alarmas.

La alta variabilidad implica que, en determinados momentos, sea conveniente evaluar el estado operativo de una grúa horquilla en el edge, mientras que en otros momentos resulte mejor hacerlo en el cloud.

Caso de Estudio

↘

Generación del Dataset

1. **Mediciones preliminares Toyota 8FG45N**
   - Acelerómetro PCE-VDL 16I.
   - Frecuencias de 25Hz.
2. **Recolección de datos.**
   - IMU BMI270 en nodo sensor.
   - Ventanas de 10s a 50Hz.
   - Datos guardados en microSD.
3. **Etiquetado de datos.**
   - Estandares OSHA.
   - Observaciones operadores.
4. **Particionado del dataset.**
   - Total: 25.000 secuencias de 500 muestras.
   - 60% Train, 20% Val y 20% Test.

Columna	Contenido
acc_x	Aceleración eje X.
acc_y	Aceleración eje Y.
acc_z	Aceleración eje Z.
gyr_x	Velocidad angular eje X.
gyr_y	Velocidad angular eje Y.
gyr_z	Velocidad angular eje Z.
label	*Good (0), Acceptable (1), Unsatisfactory (2), and Unacceptable (3)*

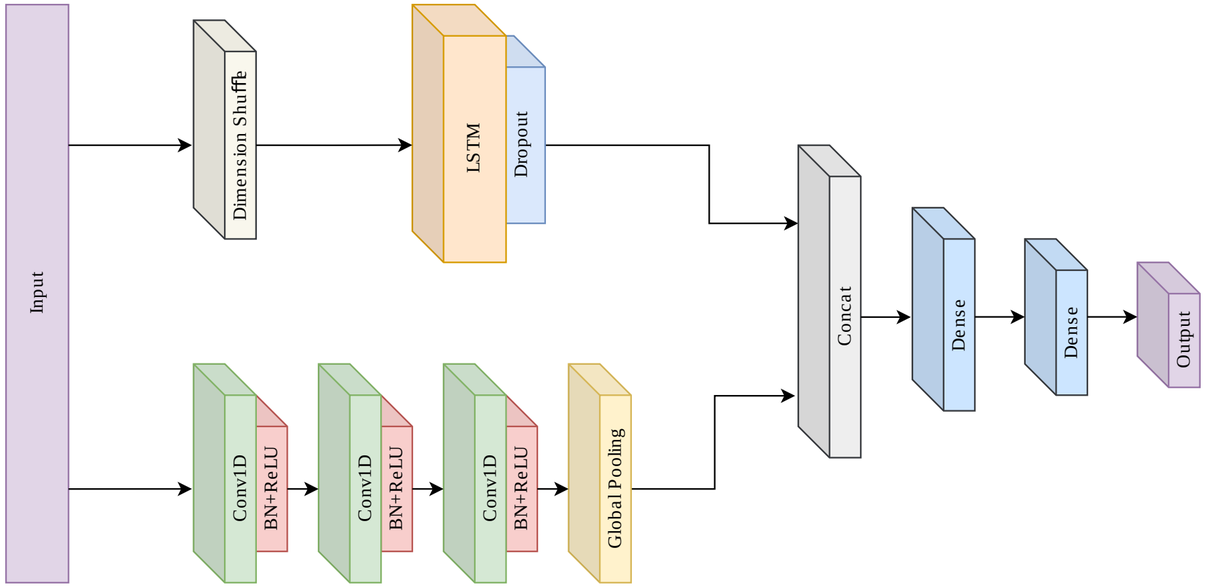
Dataset Structure

Caso de Estudio

↘

Preparación del Modelo DL

- Variante modelo LSTM-FCN propuesto por Karim et al. [13].
- Clasificador de señales digitales estudiado a detalle en escenarios.



Versión del Modelo	Tipo de Capa	Numero de Unidades
Large	LSTM Conv1D Dense	(64) (128, 256, 128) (128, 64)
Medium	LSTM Conv1D Dense	(32) (64, 128, 64) (64, 32)
Tiny	LSTM Conv1D Dense	(16) (32, 64, 32) (32, 16)

Model Architecture Versions

Caso de Estudio

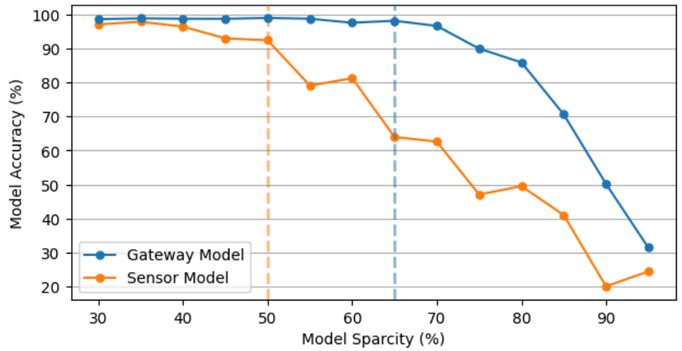
↘

Optimización usando TinyML

- **Prunning:**
  - *Medium:* 65% sparcity final.
  - *Tiny:* 50% sparcity final.

- **Quantization:**
  - *Medium:* dynamic-range.
  - *Tiny:* full-integer.

Model Accuracy Evolution During Prunning.



PdM Model	Total parameters	Size (KB)	Gzipped Size (KB)
Cloud Model	252,868	3,032	2,755
Gateway Model	64,996	90	56
Sensor Model	17,140	30	16

Model Optimization Summary.

Caso de Estudio

↘

Desempeño del Clasificador

- Se compararon los desempeños de los modelos utilizando las metricas de accuracy y recall.
- Todos los modelos obtuvieron un accuracy superior al 90%.

PdM Model	Accuracy (%)	Recall Good class (%)	Recall Acceptable class (%)	Recall Unsatisfactory class (%)	Recall Unacceptable class (%)
Cloud Model	99.38	98.11	99.71	98.56	99.69
Gateway Model	94.06	84.40	96.77	98.09	95.11
Sensor Model	91.40	81.13	99.41	97.81	99.61

Evaluación Experimental

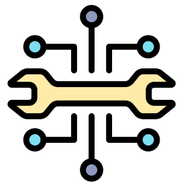
Evaluación Experimental

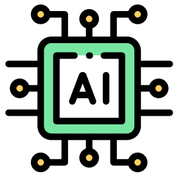
↘

- Se evaluó el desempeño del ESN-PdM Framework bajo los **3 modos de inferencia** usando las siguientes métricas:
  - Latencia de inferencia.
  - Consumo energético del nodo y vida útil batería.
  - Trafico en la WSN *(throughput)*.

- Ademas, se estudio el comportamiento de estas variables cuando el **mecanismo adaptativo** esta **activado**.







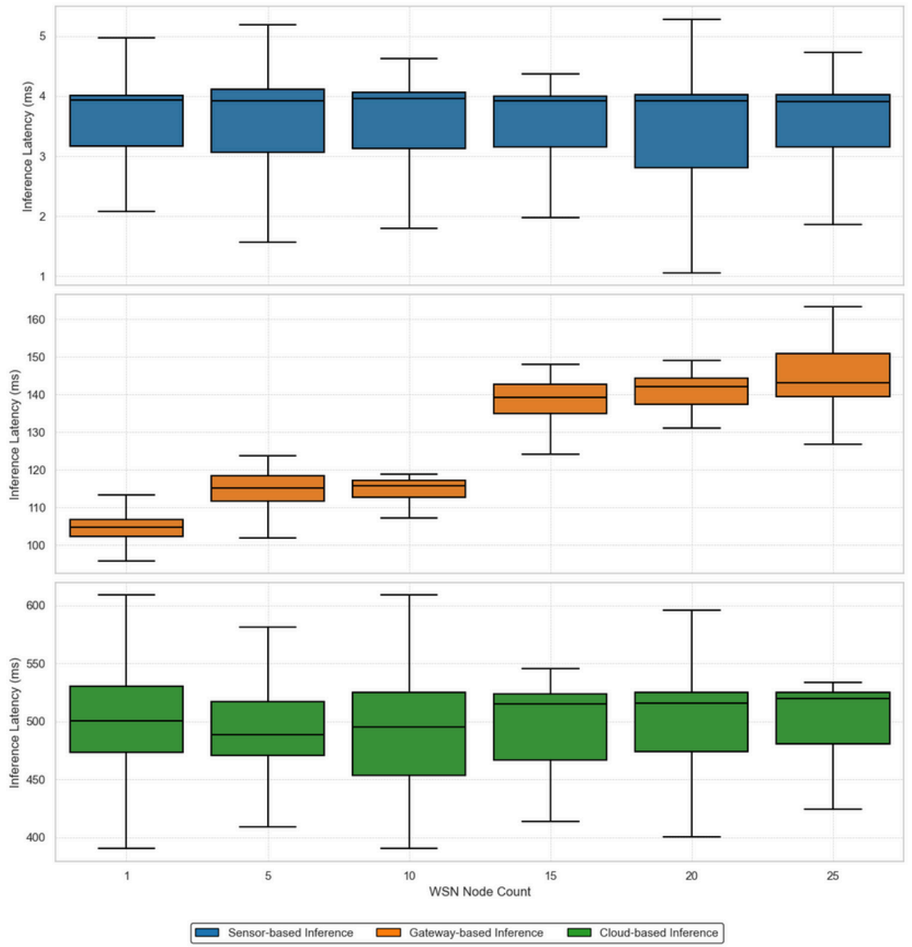
Latencia de Inferencia

Tiempo transcurrido desde que un nodo solicita una predicción hasta el momento que recibe una respuesta.

↘

Inferencia Tradicional

- Se midió la latencia al escalar la WSN desde 1 hasta 25 nodos utilizando **nodos virtuales**.
- **Sensor:** Latencia promedio de 4 ms, estable al incremento de nodos en la WSN.
- **Gateway:** Latencia promedio inicial de 105 ms, sube a 140 ms con el escalado.
- **Cloud:** Latencia promedio de 500 ms oscilando ± 25ms, estable al escalado.



Latency Over WSN Node Count

Latencia de Inferencia

↘

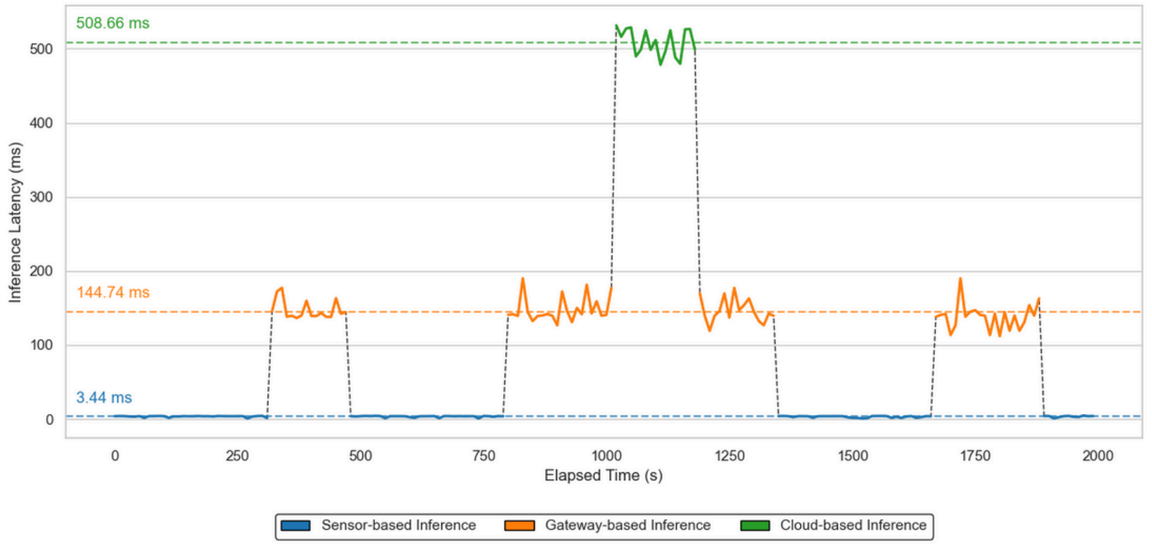
Inferencia Adaptativa

- Se considero el escenario donde la WSN cuenta con 25 nodos.
- Se midió la latencia para un nodo durante 30 minutos, durmiendo 10s entre cada medición.
- El nodo virtual se configuró con un 30% de probabilidad de emitir una anomalía.

Configuración de Parámetros:

$h_s = 32$	$h_g = 16$	$h_c = 8$
$\psi_s = 4$	$\psi_g = 8$	—
—	$\phi_g = 4$	$\phi_c = 2$

La confiabilidad de los modelos aumenta en capas superiores; incrementar ψ y reducir h permite distinguir entre anomalías reales y falsos positivos.



Latency Over Time When the Adaptive Mechanism is Enabled

Consumo Energético

Energía total consumida por un nodo durante un ciclo completo de operación, considerando tanto las fases activas como las inactivas.

↘

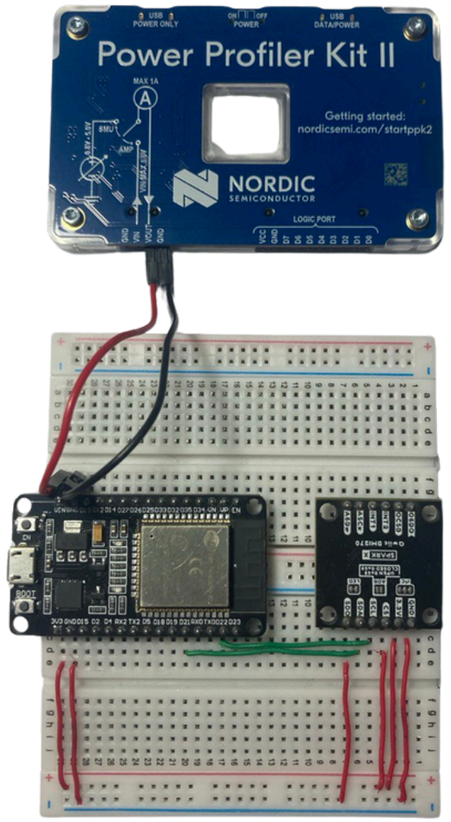
Inferencia Tradicional

- Según el modo, el nodo realiza la inferencia localmente (*onboard*) o la delega a una capa superior (*offboard*).

- Los perfiles de corriente en ambos casos fueron estudiados usando un *Power Profiler* Kit 2 de *Nordic Semiconductor*.

- **Onboard Inference:**
  1. Recolectar datos.
  2. Inferencia usando TFLM.

- **Offboard Inference:**
  1. Recolectar datos.
  2. Comprimir señal.
  3. Transmitir datos.

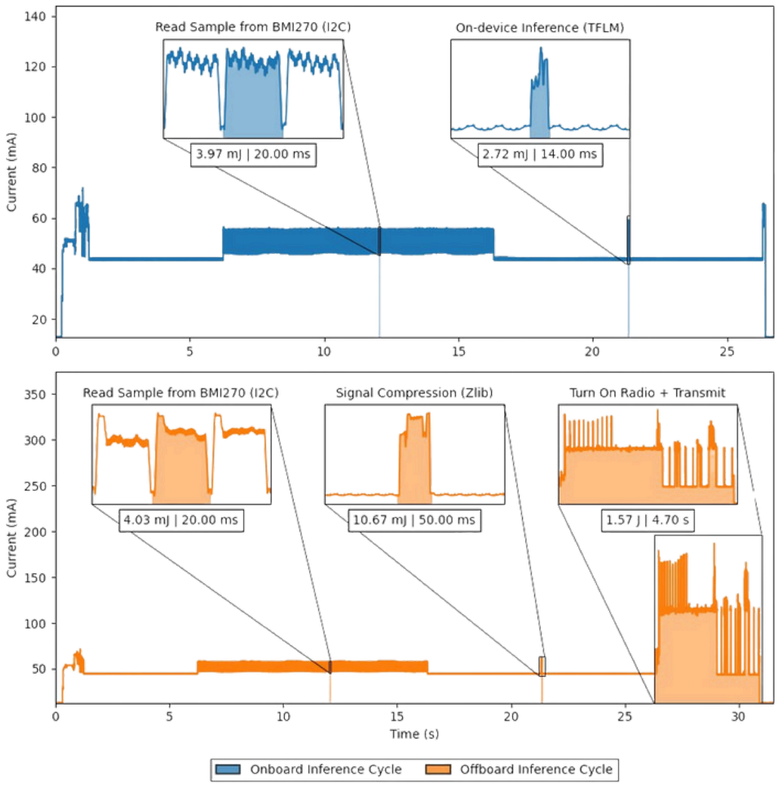


Experimental Setup

Consumo Energético

↘

Inferencia Tradicional



Current Profiles for Onboard & Offboard Inference Cycles.

- Operaciones separadas por 10 segundos.
- Consumo energético de cada operación fue estimado con:

$$E = \int_{t_0}^{t_f} V(t)I(t)\,dt$$

Donde el voltaje es 3.7 V constante, dado por la bateria LiPo.

Measured Operation	Data Size (KB)	Duration (ms)	Energy (mJ)
Data Sampling	5.86	10000	2000.00
TFLM Inference	2.92	14	2.72
Signal Compression	5.86	50	10.67
Radio Transmission	3.00	4700	1570.00

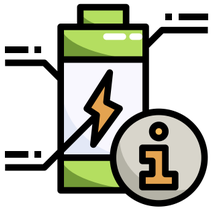
Energy Consumption per Measured Operation.

Consumo Energético

↘

Inferencia Tradicional

- Ciclos de **inferencia local** consumen *2.0 J* durante *40 s*.
- Ciclos de **inferencia remota** consumen *3.6 J* durante *45 s*.
- Batería LiPo de 1400mAh a 3.7 V ofrece 18.6 kJ de energía.
- Se estima la vida útil de la batería con la siguiente expresión:



$$\text{Battery Life} = \frac{E_{\text{battery}}}{E_{\text{cycle}}} \times t_{\text{cycle}}$$

- Realizar **únicamente inferencia remota** proporciona *65 horas* de batería, mientras que la **inferencia local** ofrece *103 horas*.

Consumo Energético

↘

Inferencia Adaptativa

- Para que se consuma toda la batería:

$$E_{\text{battery}} = n_{\text{on}} E_{\text{on}} + n_{\text{off}} E_{\text{off}}$$

$$\Rightarrow n_{\text{off}} = \frac{E_b - n_{\text{on}} E_{\text{on}}}{E_{\text{off}}}$$

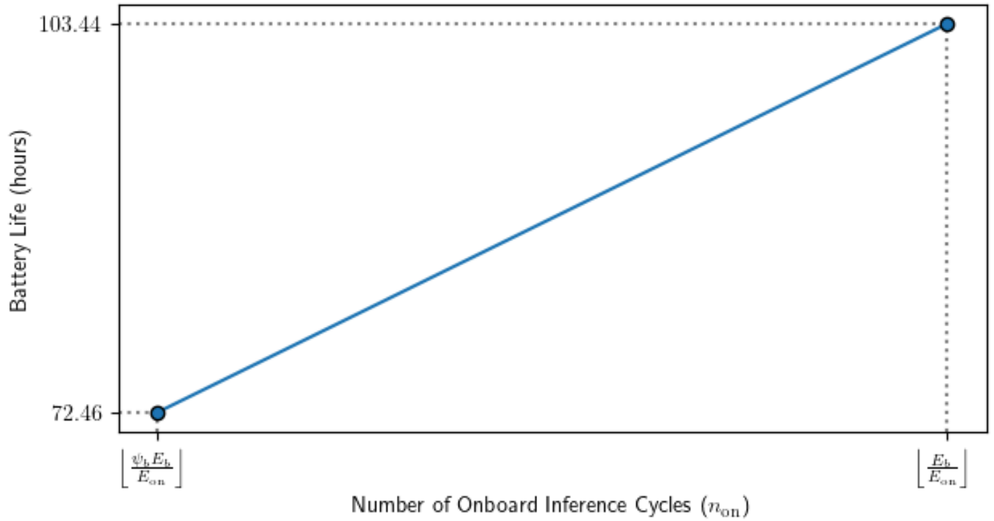
- La vida útil de la batería es:

$$\text{Battery Life}(n_{\text{on}}, n_{\text{off}}) = t_{\text{on}}\, n_{\text{on}} + t_{\text{off}}\, n_{\text{off}}$$

$$\Rightarrow \text{Battery Life}(n_{\text{on}}) = n_{\text{on}} \left(t_{\text{on}} - \frac{E_{\text{on}} t_{\text{off}}}{E_{\text{off}}} \right) + \frac{E_b t_{\text{off}}}{E_{\text{off}}}$$

- Nodos vuelven al modo sensor cuando la batería restante es menor a ψ_b.

$$n_{\text{on}} \in \left\{ \left\lfloor \frac{\psi_b E_b}{E_{\text{on}}} \right\rfloor, \ldots, \left\lfloor \frac{E_b}{E_{\text{on}}} \right\rfloor \right\}$$



Potential Battery Lives when Adaptive Mechanism is Enabled.

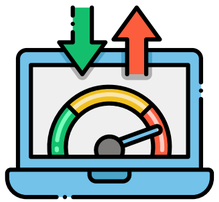
Throughput

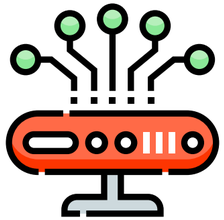
Tasa de datos transmitidos exitosamente por unidad de tiempo, es una métrica eficaz para medir el tráfico en una WSN.

↘

Inferencia Tradicional

- Escenario considera la WSN con **25 nodos virtuales**, cada uno generando una medición cada *30 s*.
- El caso de inferencia en sensor **no** será estudiado ya que el **trafico es nulo**.
- En cada caso, se capturaron los paquetes de entrada y salida del gateway con *Wireshark*.
- La tasa se calculó sumando los bits en tránsito en ventanas de 1 s.





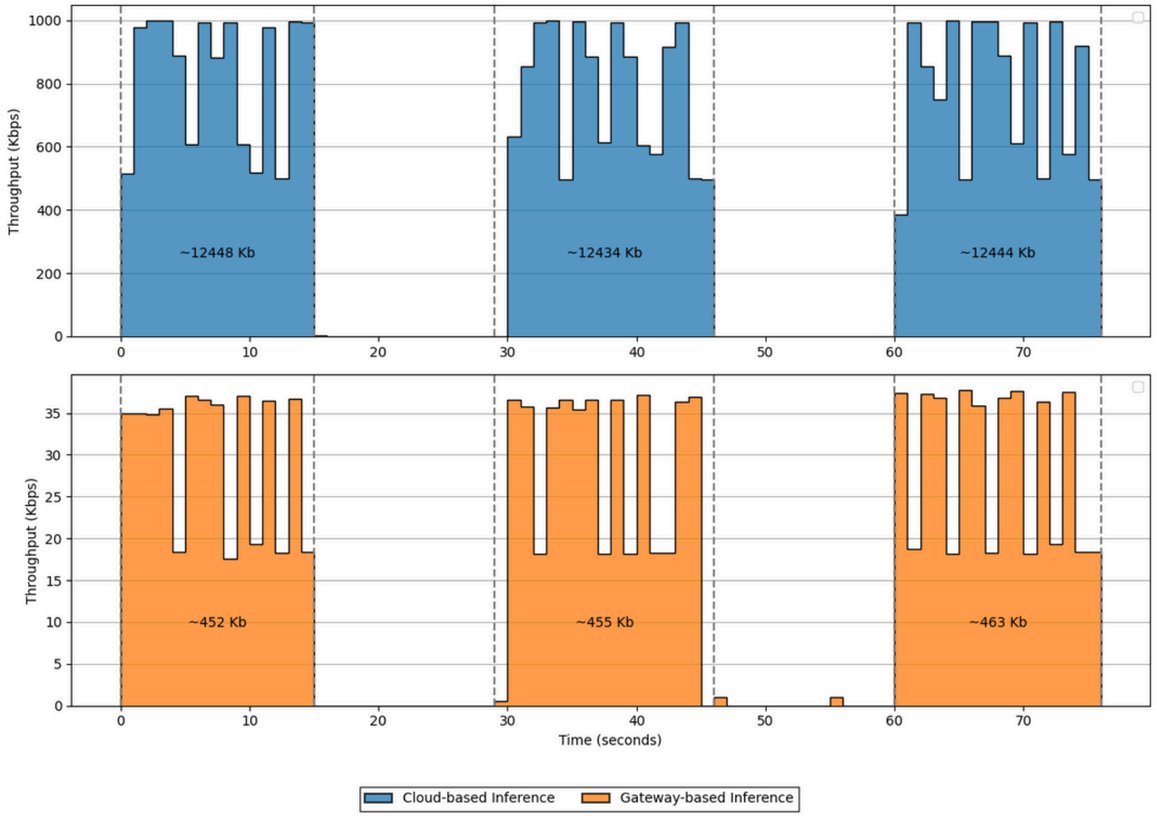
Throughput

↘

Inferencia Tradicional

- **Gateway:**
  - Oscila entre *17 Kbps* y *35 Kbps*.
  - Se transmiten *~455 Kb*.
- **Cloud:**
  - Oscila entre *500 Kbps* y *1 Mbps*.
  - Se transmiten *~12 Mb*.

$$\text{Secuencia} \Rightarrow 5.85\,\text{KB}.$$
$$\text{Tasa de compresion } 3.5:1 \Rightarrow 1.67\,\text{KB}.$$
$$\text{Base64 aumenta tamaño en } 33\% \Rightarrow 2.22\,\text{KB}.$$
$$\text{Nodos envian } 55.5\,\text{KB en total} \Rightarrow\approx 450\,\text{Kb}.$$
$$\text{Gateway envia datos como JSON} \Rightarrow\approx 12\,\text{Mb}.$$



Throughput over Time.

Throughput

↘

Inferencia Adaptativa

n_g : Numero de nodos en modo `GATEWAY`.

n_c : Numero de nodos en modo `CLOUD`.

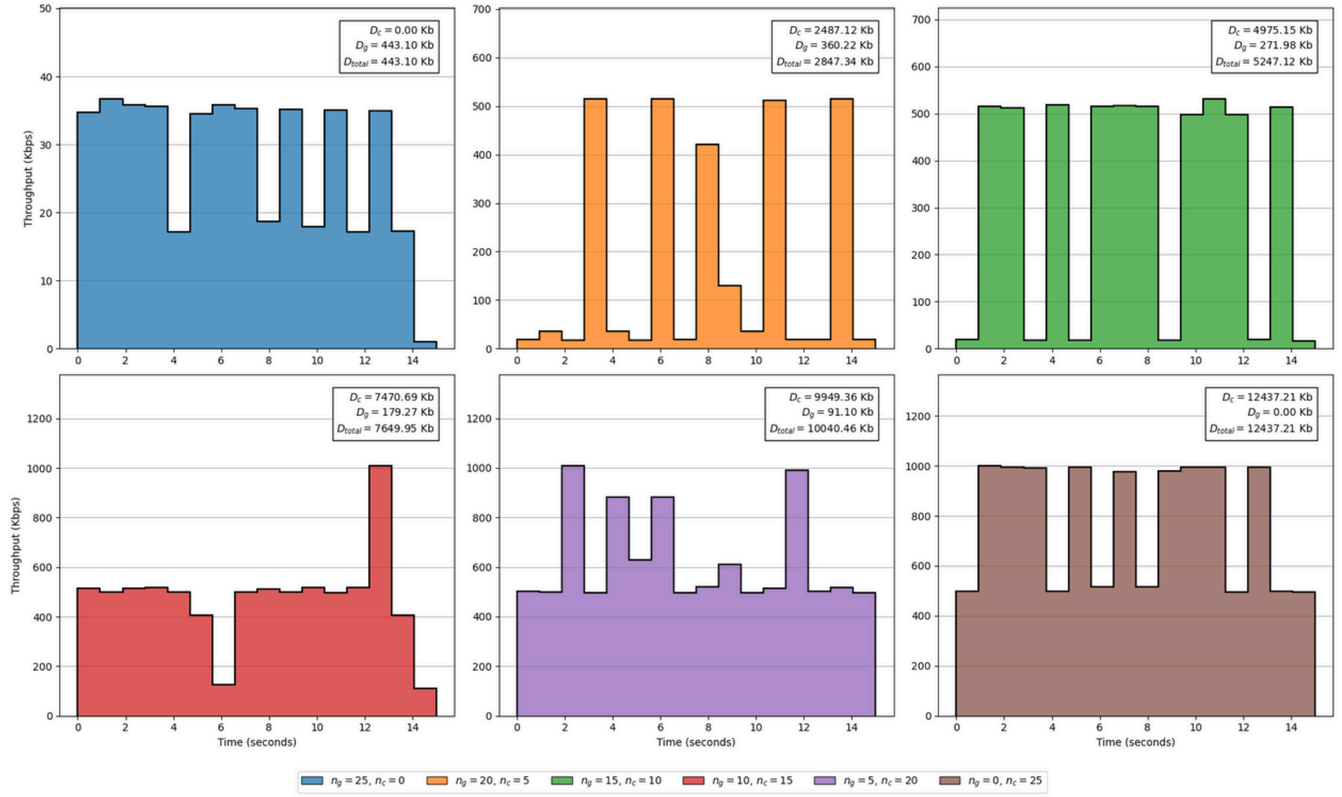
$n_{total} = 25 = n_g + n_c$

D_g : Datos transmitidos por nodos `GATEWAY`.

D_c : Datos transmitidos por nodos `CLOUD`.

$D_{total} = D_g + D_c$

- Throughput es dominado por aquellos nodos en modo Cloud.
- Contribución nodos en modo Gateway es despreciable a partir de $n_c \geq 15$.



Throughput over Time as Nodes Transition from Gateway to Cloud Inference.

Conclusiones

Conclusiones

↘

ESN-PdM

- Un framework de mantenimiento predictivo ideal para entornos altamente fluctuantes como la minería.
- Integración de enfoques tradicionales en una sola solución unificada.
- Capacidad de definir dónde ocurre la inferencia en cada nodo, manualmente o automáticamente.

Contribuciones

- Una versión *alpha* de la propuesta, completamente open-source (MIT License).
- Cuantificación de los *trade-offs* asociados a la ubicación del ML.
- Un articulo científico por publicar en el *journal IEEE Access*.

Trabajo Futuro

- Descubrimiento de microservicios: *Consul*.
- Mejorar seguridad microservicios: *Hashicorp's Vault*.
- *Reinforcement Learning* en el mecanismo adaptativo.

ESN-PdM Framework:

A TinyML-Driven IoT System for Condition Monitoring in Non-Stationary Mining Machinery

13/11 2024

Raúl de la Fuente