SISTEMA DE GESTIÓN DE RED PARA INTERNET DE LAS COSAS

MIGUEL ANDRÉS DURÁN DAJUD

Trabajo de profundización para optar por el título de Magister en Ingeniería Electrónica

DIRECTOR

Ing. Luis Carlos Trujillo Arboleda, M.SC

PONTIFICIA UNERSIDAD JAVERIANA

FACULTA DE INGENIERÍA

DEPARTAMENTO DE ELECTRÓNICA

BOGOTÁ D.C.

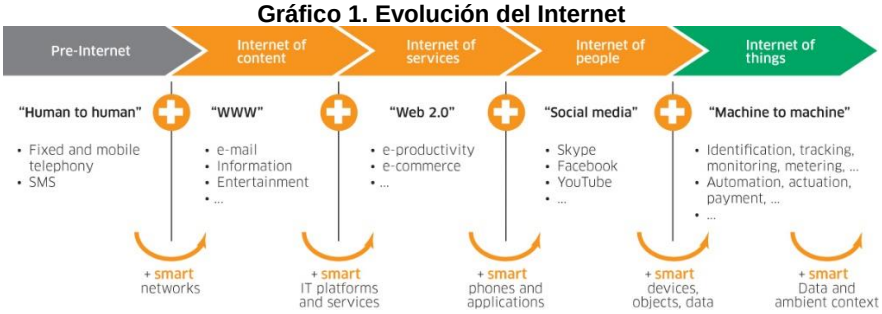
MAYO DE 2019

TABLA DE CONTENIDO

1 INTRODUCCIÓN

En la última década, Internet ha evolucionado de ser un repositorio estático de documentos hipertextuales interconectados, a un universo dinámico de personas, aplicaciones y máquinas en red [1]. Dicha evolución se presenta en el Gráfico 1. Actualmente Internet atraviesa por una nueva era conocida como el "Internet de las Cosas" -IoT-, que, de acuerdo con la Unión Internacional de Telecomunicaciones, se define como una "*infraestructura mundial para la sociedad de la información que propicia la prestación de servicios avanzados mediante la interconexión de objeto (físicos y virtuales) gracias a la interoperatividad de tecnologías de la información y las comunicaciones presentes y futuras*" [2].

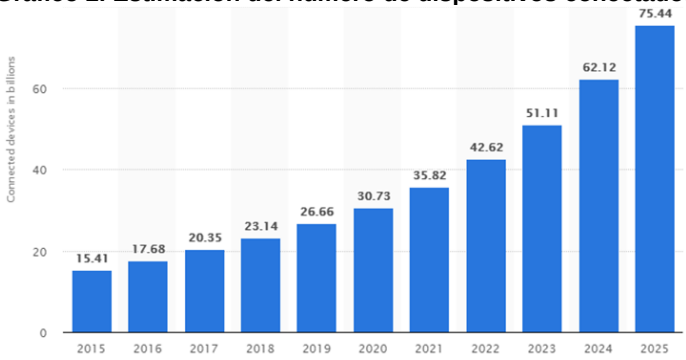
Gráfico 1. Evolución del Internet



Fuente: Tomado de [1]

De acuerdo con Ericsson. [3], en 2018 fueron utilizados 8,6 mil millones de objetos conectados, cifra que llegará a 22,3 mil millones en 2020. En [4], Statista estima un crecimiento similar, y presenta las cifras año a año de número de dispositivos conectados (ver Gráfico 2). Por su parte, las cifras estimadas por Cisco [5] predicen un mayor crecimiento, ya que calculan que para el año 2020 habrá 50.000 millones de dispositivos conectados.

Gráfico 2. Estimación del número de dispositivos conectados



Fuente: Tomado de [4]

De acuerdo con Cisco [6], para 2020, los datos producidos por las personas, las máquinas y las aplicaciones (Internet of Everything) alcanzará los 600 zeta bytes por año. Sin embargo, el tráfico global de los centros de datos solo alcanzará los 15,3 zeta bytes para esa misma fecha.

Debido a la gran cantidad de dispositivos conectados a la red, IoT supone un gran reto en términos de gestión de red, sumado a que muchas veces estos dispositivos no cuentan con capacidad para el procesamiento y análisis de los datos que generan. Adicionalmente, muchas aplicaciones de IoT requieren de tiempos de respuesta bajos y otras requieren de altas tasas de transmisión dada la gran cantidad de datos que producen.

Por lo anterior, ha surgido un nuevo paradigma en IoT denominado Fog Computing, Edge Computing, o en español, computación en el borde, en donde un gran número de dispositivos heterogéneos (inalámbricos y a veces autónomos), ubicuos y descentralizados se comunican y potencialmente cooperan entre ellos y con la red, para realizar tareas de almacenamiento y procesamiento sin la intervención de la nube [7].

En el modelo de computación en el borde, un elemento importante es el Gateway IoT, pues este es quien entrega las capacidades de procesamiento y de conexión que necesitan los dispositivos IoT en el borde de la red, y que estos requieren tradicionalmente de la nube. Así mismo, es el encargado de realizar la comunicación con la nube, decidiendo qué datos deben ser enviados a nivel central [8].

Sin embargo, de acuerdo con Vaquero [7], a pesar de que existe mucho interés en el modelo de computación en el borde, aún existen retos que deben ser resueltos, los cuales están relacionados con la sincronización de las aplicaciones de los dispositivos IoT, la limitación de procesamiento/almacenamiento, la gestión de la red IoT, seguridad, estandarización, monetización y programabilidad de las aplicaciones.

El reto en términos de gestión de red está relacionado con que en el corto plazo van a existir una gran cantidad de dispositivos IoT conectados, muchos de los cuales tienen capacidades limitadas de procesamiento, de almacenamiento y de conexión. Estos dispositivos van a requerir ser configurados y administrados una vez sean desplegados, con el fin de que las aplicaciones que se ejecutan en ellos estén correctamente configuradas y actualizadas para funcionar de forma adecuada. También es importante indicar que los dispositivos pueden tener características heterogéneas, lo que hace más complejo su administración, pues pueden requerir de configuración específicas.

Teniendo en cuenta lo anterior, el presente trabajo de grado consiste en el diseño de un sistema de gestión de red bajo el modelo de computación en el borde para una red de IoT, que permita realizar su monitoreo y configuración de manera eficiente. Así mismo, incluye la implementación de un prototipo del sistema, que permita validar el diseño realizado

Con base en lo anterior, en el presente documento se presenta la propuesta de diseño desarrollada, para lo cual, en la primera sección de abordan la generalidad de los modelos de gestión de red existentes. Con el contexto anterior, se realiza la especificación y diseño de la arquitectura del sistema de gestión propuesta, basados en el modelo de computación en el borde, realizando la selección del modelo gestión de red, dentro de los abordados en la sección anterior. En la sección 4 se explica el prototipo implementado para validar el diseño, haciendo referencia tanto al componente de software como de hardware. Así mismo, en la sección 5 se presentan las pruebas realizadas sobre el prototipo implementado. Finalmente, en la sección 6 se presentan las conclusiones del trabajo desarrollo.

2 MARCO TEÓRICO

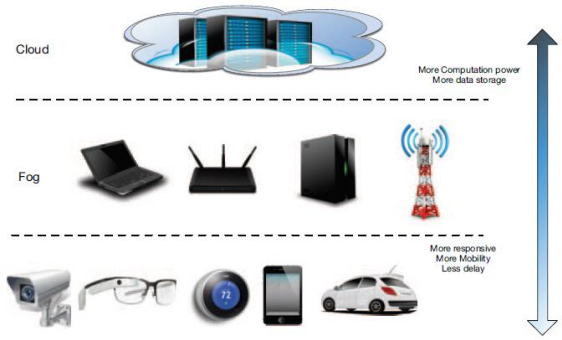
2.1 GENERALIDADES DEL MODELO DE COMPUTACIÓN EN EL BORDE

La computación en el borde es un nuevo modelo de implementación del Internet de las Cosas, en el cual el procesamiento de los datos ocurre en el punto de generación, es decir, cerca de los dispositivos IoT, y sólo se transmite la información de resumen hacia el siguiente nivel.

Este genera beneficios en términos de reducción de la tasa de transmisión utilizada para comunicarse con la nube, reducción de la latencia de las aplicaciones y mejora de la seguridad de los datos.

El modelo de computación en el borde incluye tres componentes [8]: los dispositivos IoT, que son los que generan la información, los Gateway IoT, encargados de realizar las tareas de procesamiento, almacenamiento y conexión requeridas por los dispositivos IoT, y la nube, encargada principalmente de labores de almacenamiento a largo plazo. Este modelo se muestra en el Gráfico 3

Gráfico 3. Fog Computing



Fuente: Tomado de [8]

Entre las principales tendencias en aplicaciones bajo el modelo de computación en el borde están las siguientes:

- **Sistema de semáforo inteligente (STL) [9]:** en este tipo de sistemas se despliega un STL en cada intersección de vías, con el fin de realizar tres principales acciones: (a) prevención de accidentes, a través del envío de alertas a los vehículos (vehículos conectados), (b) mantenimiento de un flujo constante de tráfico, a través de la toma decisiones relacionadas con el ciclo de los semáforos, (c) recolección de datos relevantes para evaluar y mejorar el sistema. Todas estas acciones son realizadas por los Gateway IoT.

- **Red inalámbrica de sensores y actuadores (WSAN) [10]:** El Gateway IoT lee la información de los sensores, realiza el análisis de la misma y envía acciones a los actuadores (interacción M2M[1]), sin necesidad de intervención de la nube, a la cual solo se envían ciertos tipos de reportes (interacción HMI[2]).

- **Cuidado de la salud [11]:** monitoreo de pacientes, donde se requieren tiempos de respuesta muy bajos. En estas aplicaciones se requiere que el procesamiento deba ser realizado lo más cercano al paciente.

- **Realidad aumentada [11]:** este tipo de aplicaciones son altamente intolerantes a la latencia, ya que pequeños retardos pueden dañar la experiencia del usuario. Muy utilizado en temas de juegos.

- **Identificación de rostros [12]:** el Gateway IoT detecta el rostro de una imagen tomada con una cámara, la procesa y la envía a la nube, en la cual se realiza la identificación y se envía la respuesta.

- **Analítica de video [13]:** la computación en la nube no es adecuada para aplicaciones que requieren análisis de video, debido a cuestiones relacionadas con los retardos en la transmisión de datos y temas de privacidad. Un ejemplo de esto es el relacionado con la búsqueda de

[1] Machine to Machine
[2] Human-Machine Interface

personas, en la cual los Gateway IoT realizan la búsqueda, basada en una solicitud enviada por la nube, evitando así enviarle el video.

- **Casa inteligente [13]:** en los hogares no solo se requiere que los dispositivos como las luces inteligentes, Smart TV, aspiradoras, etc., estén conectadas, sino que se requiere el despliegue de sensores y actuadores. Todos estos dispositivos generan una gran cantidad de datos y en consideración no solo al ancho de banda requerido para transportarlos, sino por temas de privacidad, estos datos deben ser consumidos directamente en el hogar.

- **Red eléctrica inteligente:** con la computación en el borde, los sistemas SCADA puede ser complementados con un modelo descentralizado de micro-grids, con lo cual no solo se mejora la escalabilidad, eficiencia de costo, seguridad y respuesta rápida del sistema, sino que también permite integrar los generadores de potencia distribuidos a la red principal.

2.2 GESTIÓN DE REDES

La gestión de red es entendida como "*la planificación, la organización, la supervisión y el control de elementos de comunicaciones para garantizar un adecuado nivel de servicio, y de acuerdo con un determinado coste*" [14]. Su principal objetivo es mejorar la disponibilidad, el rendimiento y la efectividad de los elementos de la red.

La gestión de red está basada en el paradigma gestor-agente, en el cual el primero ejecuta aplicaciones que supervisan y controlan permanentemente los elementos administrados de la red, y el segundo, se encuentra ubicado en los elementos de red y ejecuta las acciones invocadas por el gestor [14].

De acuerdo con Martí [14], independiente del modelo de gestión implementado, un sistema de gestión de red consta de las siguientes áreas funcionales:

- Gestión de fallos: comprende el conjunto de facilidades que permiten realizar la detección, el aislamiento y la corrección de fallos, así como la corrección de la operación anormal.

- Gestión de contabilidad: permite establecer cargos por el uso de recursos, e identificar costos correspondientes a la utilización de esos recursos.

- Gestión de configuración: permiten identificar, controlar, recoger y proporcionar datos a objetos gestionados, con el fin de asistir a la operación de servicios de interconexión.

- Gestión de funcionamiento: permite evaluar el comportamiento de recursos y la efectividad de actividades de comunicación.

- Gestión de seguridad: permite soportar la aplicación de políticas de seguridad.

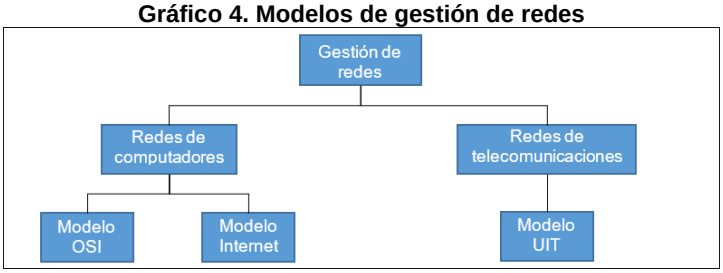
Partiendo de lo anterior, existen 3 principales modelos de gestión de redes:

- Gestión de Sistemas OSI, definido en las recomendaciones UIT-T de la serie X.700.

- Red de Gestión de las Telecomunicaciones -TMN-, definido en las recomendaciones UIT-T de la serie M.3000.

- Red de Internet, definido en varias recomendaciones de la IEFT[3], asociadas con la familia de protocolos SNMP[4].

Gráfico 4. Modelos de gestión de redes



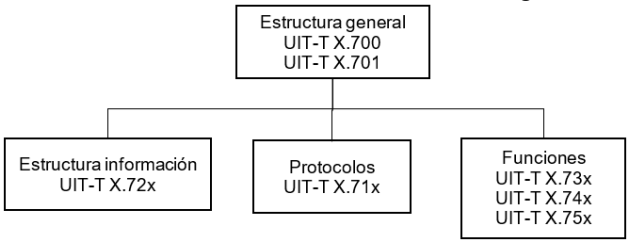
Fuente: Tomado de [16]

A continuación, se realiza una breve descripción de cada uno de los modelos de gestión mencionados.

2.2.1 MODELO DE GESTIÓN OSI

Este modelo, denominado gestión de sistemas OSI, está definido en las recomendaciones UIT-T de la serie X.700 [15], y su fundamento es la base de datos que contiene la información de los elementos gestionados. En OSI, cada recurso que se monitoriza y controla está representado por un objeto gestionado, al cual se traslada la complejidad de la gestión.

Gráfico 5. Estructura de normas del modelo de gestión OSI



Tomado de [16]

En la gestión de sistemas OSI se definen 4 aspectos principales:

- **Aspectos de información (Recomendación UIT-T X.720):** Se definen los objetos gestionados, sus atributos, las operaciones de gestión que pueden realizarse en ellos, y las notificaciones que pueden emitir. El conjunto de objetos gestionados en un sistema junto con sus atributos constituye la base de información de gestión - MIB[5].

  Un objeto gestionado es la visión de gestión de OSI de un recurso que está sujeto a gestión. Es la abstracción de este recurso que representa sus propiedades vistas por la gestión. Pueden ser de 2 tipos:

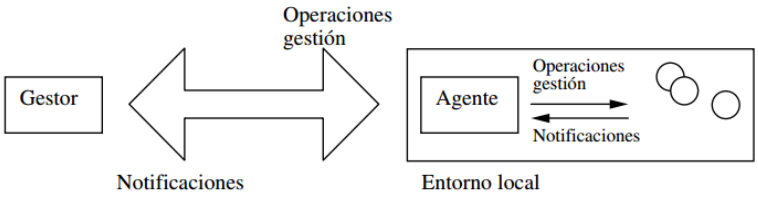
  - Objetos gestionados de capa (N), cuando son específicos de una capa determinada del modelo OSI.
  - Objetos gestionados de sistemas, que pertenecen a más de una capa, a una función específica de gestión de sistemas (objeto de soporte de gestión) o al sistema en su totalidad.

[3] Internet Engineering Task Force
[4] Simple Network Management Protocol
[5] Management Information Base

Los objetos gestionados tienen atributos, los cuales son sus propiedades, y tienen un valor asociado, que puede tener una estructura simple o compleja. Una parte importante del objeto es la especificación del conjunto de operaciones de gestión que pueden realizarse en el mismo y el efecto que estas operaciones de gestión tienen sobre el objeto gestionado y sus atributos.

Los objetos gestionados pueden enviar notificaciones, que contienen información relacionada con la ocurrencia de algún evento asociado al objeto gestionado.

Gráfico 6. Esquema del proceso de gestión en un entorno de red



Tomado de [14]

El modelo de información hace uso de los principios del diseño orientado a objetos, en el cual existe un árbol de registro, que sirve para que todos los elementos definidos en una MIB puedan tener asignados un identificador de objeto - OID[6].

- **Aspectos funcionales:** se definen las 5 áreas funcionales en la que se divide la gestión de red, a saber: gestión de fallos, gestión de configuración, gestión de configuración, gestión de contabilidad y gestión de seguridad.

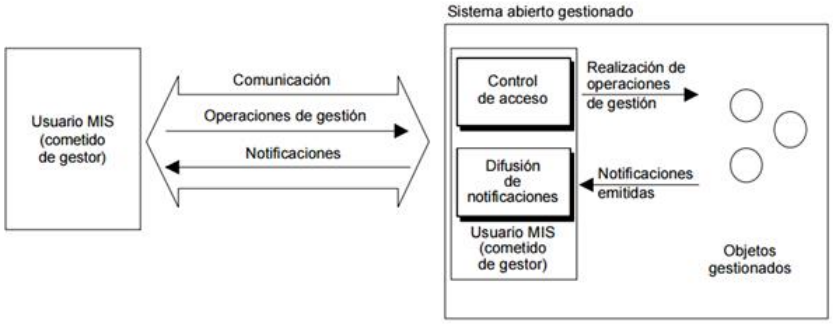
- **Aspectos de comunicaciones:** se detalla el protocolo de gestión, denominado CMIP[7], y el servicio que proporciona. Entre las características más importantes de este protocolo se destacan las siguientes [14]:

  - Requiere de gran cantidad de memoria y capacidad de CPU.
  - Se generan largas cabeceras en los mensajes.
  - Las especificaciones son difíciles de realizar y de implementar en aplicaciones.
  - La comunicación con los agentes está orientada a la conexión, por lo que asegura que los mensajes llegan a su destino.
  - La estructura de funcionamiento es distribuida.
  - Permite una jerarquía de sistemas de operación.
  - El hecho de que se trate de una gestión conducida por eventos se traduce en que existe menor gestión de tráfico, con la desventaja de tener agentes más complejos.

  Así mismo, CMIP proporciona tres tipos de servicio:
  - Manejo de datos: usado por el gestor para solicitar y alterar información de los recursos del agente.
  - Informe de sucesos: usado por el agente para informar al gestor sobre diversos sucesos de interés.
  - Control directo: usado por el gestor para solicitar la ejecución de diversas acciones en el agente.

[6] Object Identifier
[7] Common Management Information Protocol, definido en la Recomendación UIT-T X.711

Gráfico 7. Soporte de comunicación para notificaciones y operaciones de gestión



Tomado de [15]

- **Aspectos organizacionales [14]:** aquí se exponen las posibles subdivisiones de la red en dominios de gestión. Esta división se realiza a partir de políticas funcionales (por ejemplo, dominios asociados con una misma política de configuración, seguridad, contabilidad, etc.), o mediante otras políticas, como dominios geográficos (por cercanía), tecnológicos (tecnología similar o compatible), organizativas (departamentos dentro de una empresa), etc.

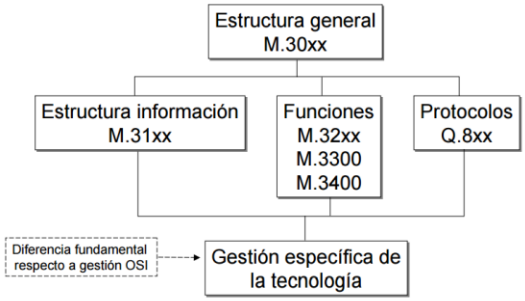
  La red se estructura en dominios administrativos, estableciendo las responsabilidades de cada dominio. Así mismo, el sistema permite que dentro de un mismo dominio, se pueda reasignar dinámicamente el labor de gestores y agentes.

2.2.2 MODELO DE GESTIÓN UIT

Este modelo, denominado Red de Gestión de las Telecomunicaciones -TMN- [17], está definido en las recomendaciones UIT-T de la serie M.3000, en la cual se proporcionan funciones de gestión y comunicaciones para la operación, la administración y el mantenimiento de una red de telecomunicaciones y sus servicios en un entorno de múltiples fabricantes. El modelo trata de hacer posible la interconexión de diferentes tipos de sistemas y equipos, permitiendo el intercambio de información de gestión mediante interfaces normalizados.

TMN fue desarrollado por la creciente heterogeneidad en la tecnología para la construcción de redes de telecomunicaciones, y la coexistencia de redes analógico-digitales. Otra razón tiene que ver con las mayores demandas sobre: posibilidad de introducir nuevos servicios, alta calidad de servicios, posibilidad de reorganizar las redes. métodos eficientes de trabajo para operar las redes y competencia entre empresas operadoras privadas.

Gráfico 8. Estructura de normas del modelo de gestión UIT



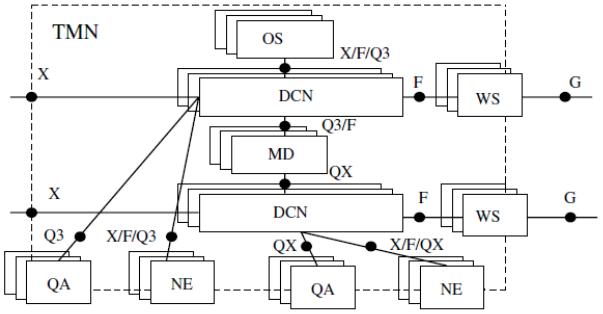
Tomado de [16]

Una TMN es paralela a la red de telecomunicaciones; se conecta con ella en ciertos puntos y controla su operación. Puede usar parte de la propia red de telecomunicaciones como soporte de sus propias comunicaciones.

TMN consta de los siguientes modelos y arquitecturas [14]:

- **Arquitectura física**, la cual proporciona la forma de transportar la información de los procesos relacionados con la gestión, a través de la definición de una serie de componentes (ver Gráfico 9):
  - Sistemas de operaciones (OS)
  - Redes de comunicaciones de datos (DCN)
  - Dispositivos mediadores (MD)
  - Estaciones de trabajo (WS)
  - Elementos de red (NE)
  - Adaptadores Q (QA).

Gráfico 9. Arquitectura física de la TMN



Tomado de [14]

Para la comunicación entre los diferentes componentes, se definen varios tipos de interfaces:
  - Interfaz Qx: utilizada para pequeños elementos de red que requieran pocas funciones operación, administración y mantenimiento, pero con gran frecuencia. Es utilizada normalmente por los NEs y MDs más complejos.
  - Interfaz Q3: esta soporta un complejo conjunto de funciones y requiere el servicio de varios protocolos para poderlas ofrecer. Está compuesta por:
    - Modelo de comunicaciones: protocolo CMIP
    - Modelo de información: semántica de datos (MIB's GDMO).
  - Interfaz X: la cual soporta el conjunto de funciones para la interconexión de diferentes OSs.
  - Interfaz F: para la interconexión de estaciones de trabajo con componentes físicos de la red de comunicaciones.

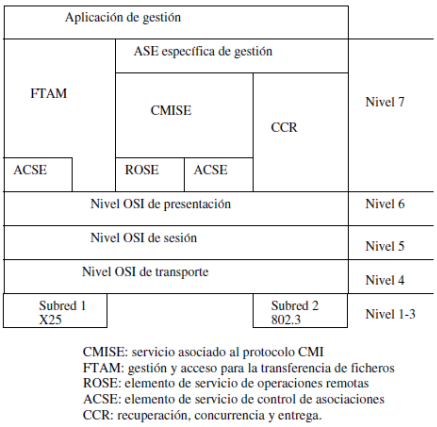
- **Modelo organizativo**, en el cual se definen las siguientes capas de gestión, que van más allá desde un punto de vista técnico:
  - Capa de gestión comercial, en la que se realiza la gestión completa y responsabilidad de empresa total, tareas de asignación de objetivos y acción ejecutiva
  - Capa de gestión de servicios, con las funciones asociadas con la interfaz con clientes y otras administraciones, la interacción con proveedores de servicios, el mantenimiento de los acuerdos de nivel de servicio, el mantenimiento de datos estadísticos y la interacción entre servicios.
  - Capa de gestión de red, en la que se realiza la provisión, cese o modificación de las capacidades de la red para el soporte de servicios a clientes y el control y coordinación de todos los elementos de la red con su ámbito y dominio.

- Capa de gestión de elementos de red, en la que se realiza el mantenimiento estadístico, control y coordinación de elementos de la red.

- **Modelo funcional**, en el cual se definen los bloques funcionales de una TMN y sus puntos de referencia. Este modelo se compone de Bloques de Servicios, Componentes y Funciones de gestión.

- **Modelo de información**, el cual es el definido por la interfaz Q3 en la semántica de datos (MIBs GDMO). En el Gráfico 10 se presenta un esquema con los niveles de estructura de protocolos de la interfaz Q3. Los niveles 1 a 3 están definidos en la Recomendación UIT-T Q.811. Para los niveles altos, se utiliza la Recomendación UIT-T Q.812 que define dos tipos de perfiles de protocolo: perfiles para servicios de tipo transacción y perfiles para servicios de tipo transferencia de archivos. En el nivel de aplicación se utiliza el protocolo CMIP.

Gráfico 10. Esquema de capas de la interfaz Q3



Tomado de [14]

2.2.3 MODELO DE GESTIÓN EN INTERNET

Este modelo está basado en la utilización del protocolo SNMP, sobre redes con pilas de protocolos TCP/IP. El marco de trabajo de este modelo está contenido esencialmente en 3 documentos [14]:

- *Structure of Management Information* (SMI): RFC1155.
- *Management Information Base* (MIB): RFC 1156, RFC 1213.
- *Simple Network Management Protocol* (SNMP): RFC 1157.

Con el objetivo de poder referenciar el recurso a ser gestionado, el modelo de gestión en Internet especifica una estructura de la información. Para ello, se utilizan los identificadores de objeto -OID[8]. Estos son una secuencia de números separados por un punto que forman un árbol, el cual está estandarizado a nivel mundial. Estos OID son los que permiten nombrar objetos mediante SNMP.

Otro elemento importante del modelo de gestión en Internet, son las bases de información de gestión MIB, las cuales son un conjunto de objetos gestionados de un recurso que se publican para ofrecer interoperabilidad de gestión. Estos objetos se organizan en grupos, según sea su temática. Actualmente existen 3 tipos de MIBs: las de tipo estándar (MIB-I, contenida en el RFC 1156, y MIB-II, contenida en el RFC 1213); las experimentales, con grupos en fase de desarrollo; y las privadas, que incorporan la información de los diversos fabricantes de equipos.

[8] Object Identifiers

Con ya se indicó anteriormente, SNMP es el protocolo de comunicación utilizado en el modelo de gestión en Internet. SNMP define una arquitectura basada en cliente-servidor, en la cual el cliente (denominado gestor de red) realiza conexiones virtuales a un servidor (denominado agente SNMP) ejecutado en un dispositivo de red remoto. Los mensajes enviados por el gestor de red a los agentes SNMP están formados de OID, junto con instrucciones respectivas para obtener o cambiar un valor.

SNMP se apoya en la estructura de protocolos TCP/IP, tal y como puede observarse en el Gráfico 11. Entre las características principales del protocolo SNMP se puede destacan las siguientes:

- Es un protocolo de gran flexibilidad y permite una gran extensibilidad a todo tipo de redes.

- La eficiencia del protocolo para la transmisión de información es baja, ya que se trata de una arquitectura basada en *polling* (interrogación), de acuerdo con una estructura de funcionamiento centralizada.

- La arquitectura basada en interrogación implica que el gestor consulte periódicamente la información del agente sobre los recursos gestionados, lo que hace que el gestor pueda ser simple, pero con la desventaja de que la gestión de tráfico puede ser importante.

- Es posible utilizar un agente proxy para gestionar sistemas propietarios.

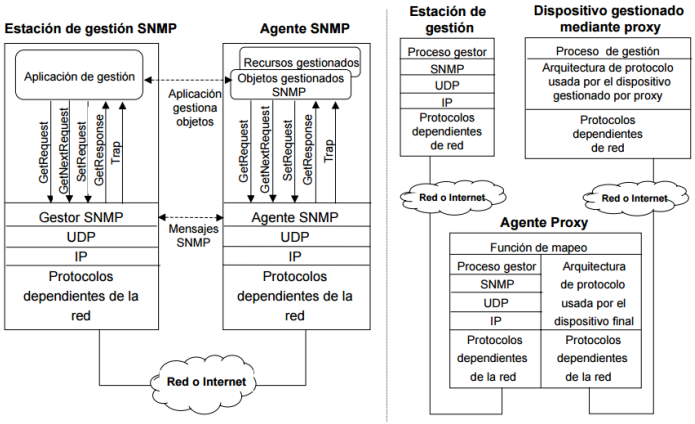
- La comunicación con los agentes no está orientada a conexión, por lo que tienen que integrarse mecanismos especiales en capas superiores para evitar estas deficiencias.

El protocolo SNMP permite realizar las siguienites operaciones:

- GetRequest: petición de valores específicos de la MIB.
- GetNextRequest: proporciona un medio para moverse por la MIB. Petición del objeto siguiente a uno dado de la MIB.
- GetResponse: entrega los valores solicitados por las operaciones anteriores.
- SetRequest: permite asignar un valor a una variable.
- Traps: permite a los agentes informar de sucesos inusuales.

Por otra parte, para realizar una gestión adecuada se define un nombre de comunidad (community), que afecta tanto al agente como al conjunto gestor, y un perfil de comunidad, que delimita las propiedades, así como el modo de acceso al sistema.

Gráfico 11. Arquitectura modelo Internet



Tomado de [16]

Teniendo en cuenta las limitaciones de SMNP, se definió una versión 2 del mismo, el cual aporta las siguientes ventajas:

- Permite una mayor eficiencia en la transferencia de información.
- Admite mecanismos de seguridad como la autentificación y el cifrado frente al SNMP (no
- implementados).
- Permite la comunicación entre estaciones de gestión.
- Parte de un modelo de comunicaciones extendido considerablemente.
- Permite una señalización extendida de errores.
- Permite el uso de varios servicios de transporte.

El protocolo SNMPv2, que incluye los mensajes de SNMPv1, permite realizar las siguientes operaciones:

- GetRequest: petición de valores específicos de la MIB.
- GetNextRequest: proporciona un medio para moverse por la MIB. Petición del objeto siguiente a uno dado de la MIB.
- GetBulkRequest: petición de múltiples valores.
- Response: devuelve los valores solicitados por las operaciones anteriores gestor a gestor o agente a gestor.
- SetRequest: permite asignar un valor a una variable.
- InformRequest: transmite información no solicitada (gestor a gestor).
- SNMPv2-Traps: permite a los agentes informar de sucesos inusuales

Así mismo, existe una versión 3 de SNMP. Sin embargo, no ha sido mayoritariamente aceptado en la industria.

2.3 MODELOS DE GESTIÓN DE REDES IOT

Los avances realizados sobre IoT han sido desarrollados por áreas específicas de la industria como la salud, el hogar, el transporte, etc., lo cual ha generado la creación de diversas arquitecturas y mecanismos de gestión, los cuales a su vez están conformados por tecnologías, protocolos y estándares diferentes.

De acuerdo con Weber [18], existen cuatro requisitos fundamentales de gestión de dispositivos para cualquier despliegue de dispositivos de IoT:

- Aprovisionamiento y autenticación: el aprovisionamiento es el proceso de inscribir un dispositivo en el sistema. La autenticación, que hace parte de ese proceso, donde sólo se registran los dispositivos que presentan las credenciales adecuadas, es el acto de establecer de forma segura la identidad del dispositivo para garantizar que se puede confiar en él, con el fin de establecer que el dispositivo es realmente un dispositivo genuino, ejecuta software confiable y está trabajando en nombre de un usuario de confianza.

- Configuración y control: La mayoría de las veces, los dispositivos deben ser configurados de acuerdo con condiciones específicas particulares de cada entorno, y con las preferencias de los usuarios. Así mismo, para implementar cierta capacidad de control en un sistema, es deseable restablecer el dispositivo de forma remota para lograr un estado conocido y recuperarse de errores e implementar nuevos cambios de configuración. También es posible que se desee restablecer el dispositivo a una configuración predeterminada de fábrica, lo cual es útil cuando se desea desactivar un dispositivo o como una forma más invasiva de recuperarse de condiciones de error desconocidas. Finalmente, enviar un comando para actualizar o volver a

cargar el firmware del dispositivo es muy importante para mantener la seguridad del mismo, implementar mejoras de características y corregir errores.

- Monitoreo y diagnóstico: En un sistema de miles de dispositivos IoT, el funcionamiento correcto de cada uno de ellos puede afectar directamente a la operación del sistema, pudiendo afectar los resultados financieros. Por lo tanto, el monitoreo y el diagnóstico son vitales para minimizar el impacto de cualquier tiempo de inactividad del dispositivo debido a errores de software u otros problemas operativos imprevistos.

- Actualizaciones de software y mantenimiento: la actualización de los dispositivos IoT es requerida debido a que el software de los mismo puede contener errores, o porque se desea adicionar características y funcionalidades, o debido a que es necesario soluciones vulnerabilidades de seguridad.

En este punto es importante tener en cuenta que, los modelos tradicionales de gestión de redes (OSI, UIT e Internet), fueron diseñados teniendo en cuenta que los dispositivos a gestionar cuentan con las capacidades suficientes, de procesamiento y de comunicación, para soportarlos.

En el mundo de IoT, esto no es así, ya que muchos de los dispositivos tienen capacidades limitadas de procesamiento, de almacenamiento y de conexión, y requieren que las aplicaciones que se ejecutan en ellos estén correctamente configuradas y actualizadas para funcionar de forma adecuada.

A partir de lo anterior, se han desarrollado diversos estudios para la implementación de la gestión de IoT. De acuerdo con Rodríguez [19], existen 4 tendencias relacionadas con las formas de implementación de la gestión de la IoT, las cuales se clasifican de acuerdo al mecanismo o estándar de gestión empleado: SNMP, TMN, WBEM[9], y PBM[10].

El estándar más atractivo para la implementación de la gestión de IoT es SNMP, debido a la masificación que ha alcanzado este estándar en los fabricantes. Durante los últimos años se han desarrollado varios estudios encaminados a adaptar este protocolo a una red de IoT.

Hui-Ping [20] presenta un método de gestión de IoT basado en el protocolo SNMP y los detalles de implementación para configurar la MIB, el agente y la aplicación de gestión. El agente almacena todos sus datos en un árbol de MIB, que contienen variables que contienen los valores que representan la información de los sensores. La aplicación de gestión puede enviar un mensaje al agente de los dispositivos IoT administrados incorporados con sensores y obtener el valor de las variables almacenadas en MIB, para conocer la información del sensor.

Por otra parte, Choi [21] presenta la implementación de 6LoWPAN-SNMP, que permite la transmisión de mensajes SNMP sobre redes 6LoWPAN[11], a través de la modificación del protocolo SNMP, con el fin de reducir la cantidad de tráfico SNMP, mediante el uso de varias técnicas:

- En primer lugar, se proponen métodos para comprimir los mensajes SNMPv1 y v2 sin ninguna modificación de las operaciones existentes del protocolo, con lo cual se reduce el tamaño de cada mensaje SNMP.

- En segundo lugar, para reducir el número de mensajes transmitidos a través de la red, se proponen nuevas operaciones del protocolo y soporte de broadcast / multicast en los motores SNMP. Combinando estas dos técnicas, es posible reducir grandes cantidades de tráfico de red,

[9] Web Based Enterprise Management
[10] Policy Based Management
[11] IPv6 over Low power Wireless Personal Area Networks

asegurando al mismo tiempo el correcto funcionamiento de SNMP. En este punto se incorporan los mensajes GETREquest PDU y StopGETRequest PDU que son enviados de forma periódica, lo que permite que el reporte solo sea solicitado una vez por el gestor.

- En tercer lugar, se propone un proxy 6LoWPAN-SNMP para asegurar compatibilidad con las versiones SNMP actuales y maximizar la eficiencia del 6LoWPAN-SNMP.

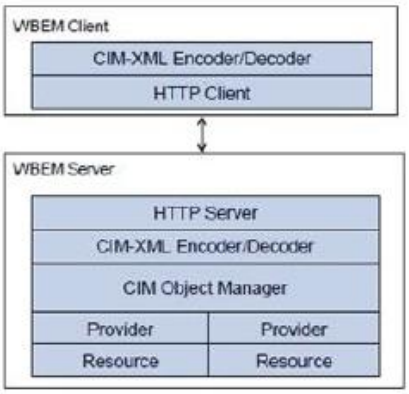
Por otra parte, según Rodríguez [19], la gestión de IoT usando TMN, tiene la característica de combinar la robustez de CMIP, con la interoperabilidad de CORBA[12], por lo que se posibilita la interoperabilidad de comunicación entre los diversos componentes de software escritos en diferentes lenguajes que van a estar integrados en los dispositivos IoT. Sin embargo, las razones fundamentales que detuvieron el desarrollo de este mecanismo en gestión son:

- No existen normas para el soporte e implementación de CORBA.
- La pila de protocolos de la arquitectura TMN es muy compleja para ser implementado en la nueva era de Internet.

De otro lado, la gestión WBEM [19], que es una iniciativa del DMTF[13], está compuesta por un conjunto de estándares y tecnologías para la gestión de Internet, desarrollados con el fin de unificar los sistemas de gestión de redes, de usuarios y aplicaciones existentes. Su principal ventaja radica en el aprovechamiento del poder de la Web para facilitar las tareas de gestión. Sus principales componentes son:

- Aplicación de gestión (WBEM-Client), la cual obtiene la información de gestión a partir de una comunicación directa con el CIMOM[14] a través de mensajes Request, en lugar de acceder directamente a los proveedores asociados a los dispositivos, lo que le brinda interoperabilidad al modelo.
- WBEM-Server, el cual generalmente es instalado en los dispositivos, y actúa como intermediario entre el gestor y el hardware del dispositivo, permitiendo ocultar los detalles de la comunicación entre ellos.

Gráfico 12. Arquitectura WBEM



Fuente: Tomado de [19]

Otros elementos importantes de la arquitectura son los protocolos empleados para la comunicación: el protocolo CIM-XML, que se encarga de la codificación/descodificación de la información, y el protocolo HTTP para el transporte de los datos.

[12] Common Object Request Broker Architecture
[13] Distributed Management Task Force, Inc
[14] Common Information Model Object Manager

Por otra parte, la gestión basada en PBM [19], consiste en hacer uso de la gestión autonómica, la cual se logra a partir del cumplimiento de las cuatro funcionalidades: auto-configuración, auto-reparación, auto-optimización y auto-protección. Por lo anterior, este tipo de gestión se destaca por modificar el rol del operador, el cual en vez de controlar el sistema directamente, realiza funciones asociadas con la definición de políticas.

Finalmente, es importante mencionar que las plataformas de servicios en la nube han desarrollado sus modelos de gestión propios. Este es el caso de Azure IoT Hub[15], el cual es un servicio administrado, hospedado en la nube, que actúa como centro de mensajes para comunicaciones bidireccionales entre la aplicación de IoT y los dispositivos que administra. Este proporciona características y un modelo de extensibilidad que permiten a los desarrolladores de back-end y de dispositivos generar soluciones sólidas de administración de dispositivos. Azure IoT Hub proporciona un conjunto de patrones de administración de los dispositivos como reinicio, restablecimiento de fábrica, configuración, actualización de firmware e informes de estado y progreso.

A la fecha no se ha desarrollado un sistema de gestión autonómico estandarizado para IoT. Únicamente se destacan algunas propuestas de software en sectores específicos del sistema de gestión autonómico.

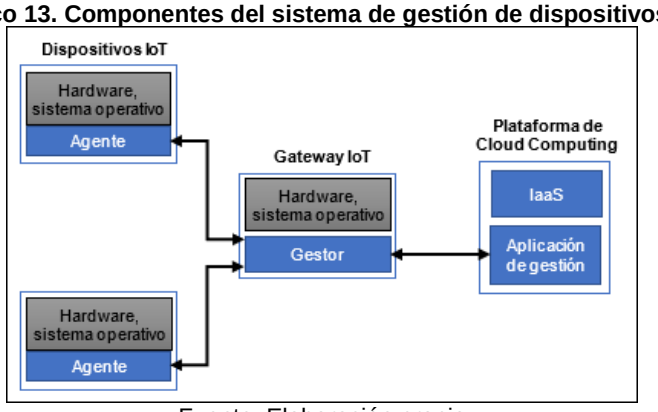
3 ESPECIFICACIÓN Y DISEÑO DE LA ARQUITECTURA DEL SISTEMA DE GESTIÓN DE RED

El objetivo del presente trabajo de grado consiste en diseñar un sistema de gestión de red, bajo el modelo de computación en el borde, para una red de IoT, que permita realizar su monitoreo y configuración, e implementar un prototipo que permita validar el diseño realizado. Con base en lo anterior, a continuación se presentan las condiciones del diseño de arquitectura y funciones de gestión.

3.1 DISEÑO DE LA ARQUITECTURA Y FUNCIONES DE GESTIÓN

El sistema de gestión de red es diseñado teniendo en cuenta los tres componentes del modelo de computación en el borde, es decir, los dispositivos IoT, el Gateway IoT y la nube (ver Gráfico 13), y bajo una arquitectura estándar de gestión de red, dentro de las que fueron identificadas en la sección 2.2 del presente documento.

Gráfico 13. Componentes del sistema de gestión de dispositivos IoT



Fuente: Elaboración propia

[15] https://docs.microsoft.com/es-es/azure/iot-hub/iot-hub-device-management-overview

Teniendo en cuenta lo anterior, el desarrollo del sistema de gestión de red implica la definición aspectos relativos a la información de los dispositivos IoT a ser gestionada, específicamente los atributos o propiedades que permitan realizar su monitoreo y configuración.

3.1.1 Selección del modelo de gestión de red a implementar

Como se explicó en la sección 2.2 del presente documento, existen 3 modelos de gestión de redes: modelo OSI, modelo UIT y modelo de Internet.

En la escogencia del modelo de gestión de red a aplicar para gestionar una red IoT, se debe tener en cuenta la limitación de los dispositivos IoT en cuanto a capacidades de procesamiento, almacenamiento y comunicación, así como la cantidad de dispositivos, el consumo energético y la tasa de transferencia.

En este sentido, en primer lugar, es importante recordar que el protocolo de comunicación utilizado en los modelos de OSI y UIT de gestión de redes es el CMIP, y el utilizado en el modelo de Internet es el SNMP. Por lo anterior, la selección se basa principalmente en la escogencia del protocolo de comunicación, y la estructura de información asociada a cada uno.

A partir de lo anterior, de acuerdo con Subramanian [22], las ventajas de SNMP sobre CMIP son las siguientes:

- Como su nombre lo indica, SNMP es realmente simple, lo que hace que sea fácil de implementar, por lo que es el sistema de gestión de red más ampliamente implementado hoy en día.
- SNMP está basado en tecnología escalar y la definición simple de objetos gestionados, lo que lo favorece sobre CMIP.
- SNMP utiliza menos memoria y recursos de procesamiento del dispositivo donde se encuentra implementado, en comparación con CMIP.

Así mismo, Luque [23] señala que SNMP se basa en un conjunto muy simple de protocolos y de estructura de información, la cual sigue el modelo orientado a objetos, pero introduce muchas simplificaciones con respecto a los modelos UIT y OSI. Esta situación hace más difícil el modelado de dispositivos, redes y servicios complejos, pero, por otro lado, facilita enormemente las aplicaciones de gestión.

Estas limitaciones del modelo de Internet son, al mismo tiempo, una ventaja y un inconveniente. Por un lado, está claro que limita las capacidades de gestión de la red. Pero, por otro lado, su diseño sencillo permite un fácil desarrollo de agentes y gestores, funciones y plataformas de gestión.

Por otro lado, las plataformas de gestión del modelo UIT soportan muchas funcionalidades y pueden describir y gestionar casi todos los aspectos particulares de una red de telecomunicaciones. Sin embargo, el número de dispositivos que son capaces de interconectarse mediante una interfaz Q3, que es la utilizada en este modelo, es muy bajo. Por esta razón, los precios de estos sistemas de gestión son notablemente más altos que sus equivalentes en SNMP.

En cuanto a las ventajas de CMIP sobre SNMP, Foreman [24], señala las siguientes:

- Las variables CMIP no sólo transmiten información, sino que también se pueden utilizar para realizar tareas, lo que no es posible en SNMP.
- CMIP es un sistema más seguro, ya que se ha construido en base a la seguridad, que admite autorización, control de acceso y registros de seguridad. SNMP no tiene implementada estas características en su versión más utilizada, que es la v2, pero sí en la v3.

- CMIP proporciona potentes capacidades que permiten a las aplicaciones de gestión realizar más tareas con una sola petición.
- CMIP proporciona un mejor reporte de las condiciones inusuales de la red.

De acuerdo con Luque [23], mientras que una MIB de SNMP siempre se puede describir de acuerdo con el modelo UIT/OSI, lo contrario no es cierto. Muchas características que modelan objetos en el modelo UIT/OSI no tienen traducción al modelo de gestión de Internet.

Por su parte, Luque [23] afirma que, en pequeñas redes con funciones de gestión limitadas, los sistemas de gestión basados en SNMP son la mejor solución, considerando el equilibrio entre precio y rendimiento. Sin embargo, afirma que, en redes de tamaño medio y grande, y cuando estas redes requieren una amplia gama de funciones de gestión, el uso de sistemas basados en SNMP puede ser un obstáculo para el presente y un riesgo para el futuro.

Es importante tener en cuenta que a través de SNMP, en principio, sólo se transmite información, y no es posible realizar tareas, lo cual afecta la función de actualización de software de los dispositivos IoT, por cuanto no permite la transferencia de archivos. Sin embargo, existen desarrollos en los cuales la solución es implementar junto con SNMP, un protocolo de transferencia de archivos. Específicamente, sobre este particular CISCO implementó una MIB[16] mediante la cual realiza la transferencia de archivos entre un router y un servidor TFTP a través de SNMP, mediante la cual se actualiza el firmware de este dispositivo.

Teniendo en cuenta todo lo anterior, en la Tabla 1 se realiza una comparación de las características de los protocolos SNMP y CMIP, con el fin de realizar la selección del modelo para la arquitecta del sistema de gestión de la red IoT.

Tabla 1. Comparación de atributos entre SNMP y CMIP

Atributo	SNMP	CMIP
Simplicidad de implementación	+	−
Consumo de recursos	−	+
Realización de tareas	+ (no de forma nativa)	+
Transferencia de archivos	+ (no de forma nativa)	+

Fuente: Elaboración propia

Teniendo en cuenta lo anterior, en el presente proyecto se seleccionó el modelo de gestión de Internet para implementar el sistema de gestión de redes IoT, teniendo en cuenta la simplicidad y la baja utilización de memoria y recursos del protocolo SNMP, en comparación con el protocolo CMIP utilizado en los modelos de gestión de OSI y UIT, lo cual es una de las características principales de los dispositivos IoT. Así mismo, a pesar de que de forma nativa SNMP no soporta la realización de tareas, existen implementaciones que permiten ejecutarlas, basadas en la definición de objetos de información desarrolladas para esta necesidad.

Por otra parte, en cuanto a la versión de SNMP, se seleccionó la versión 2c, teniendo en cuenta que es la más ampliamente implementada, y que sus funcionalidades permiten la implementación del sistema de gestión para una red IoT. No obstante, en versiones futuras es posible considerar la implementación de la versión 3, la cual entrega mejores opciones de seguridad al protocolo SNMP, tal y como puede observarse en el Gráfico 14.

[16] https://www.cisco.com/c/en/us/support/docs/ip/simple-network-management-protocol-snmp/7910-11-7910.html

Gráfico 14. Principales diferencias entre SNMP v1, v2c y v3

SNMP v1	SNMP v2c	SNMP v3
Easy to set up. Only requires a plain text community string to authenticate packets	Identical to version 1	Setup is more complex. Does not use community strings but users with authentication and encryption.
Supports only 32 bit counters	Support for 64 bit counters	Adds security to the 64 bit counters.
Packet Types: • Get-Request • Get-Next-Request • Set Request • Get Response	Packet Types: • Get-Request • Get-Bulk-Request • Get-Next-Request • Set Request • Inform-Response • SNMP v2 Trap	The basic functions of v3 are from v1 and v2. v3 has a new SNMP message format
Anybody with access to the network will be able to see the community string in plaintext	• Improved error handling • Improved SET commands	Adds both encryption and authentication, to the SNMP message.

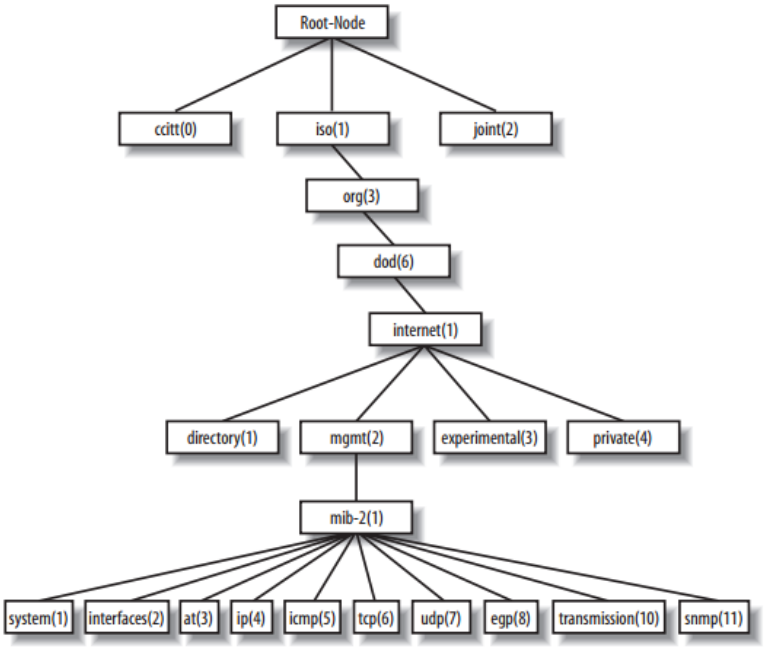
Fuente: Tomado de [25]

3.1.2 Base de datos de información

Tal y como fue explicado en la sección 2.2.3 del presente documento, el modelo de gestión en Internet especifica una estructura de la información, en la cual se utilizan OID y las bases de información de gestión MIB.

Por lo tanto, el sistema de gestión de IoT debe ser capaz de identificar las MIB de tipo estándar (MIB-II, contenida en el RFC 1213). En el Gráfico 15 se muestra el árbol de OID para SNMP, en el que se identifican los grupos de información que hace parte de las MIB-II.

Gráfico 15. Árbol de OID para SNMP



Fuente: Tomado de [26]

A través de las MIBs de tipo estándar es posible gestionar el siguiente tipo de características [26]:

- **system**: define una lista de objetos que pertenecen al funcionamiento del sistema, como el tiempo de actividad del sistema, el contacto del sistema y el nombre del sistema.
- **interface**: controla el estado de cada interfaz en un dispositivo gestionada. Monitorea qué interfaces están arriba o abajo y tiene estadísticas tales como octetos enviados y recibidos, errores y descartes, etc.
- **at** (*address translation* o traducción de direcciones): está obsoleto y sólo se proporciona para compatibilidad con la MIB-I. En él se almacenan las direcciones de nivel de enlace correspondientes a una dirección IP.
- **ip**: almacena la información relativa a la capa IP, tanto de configuración como de estadísticas.
- **icmp**: se almacenan contadores de los paquetes ICMP entrantes y salientes.
- **tcp**: información relativa a la configuración, estadísticas y estado actual del protocolo TCP.
- **udp**: información relativa a la configuración, estadísticas del protocolo UDP.
- **egp**: información relativa a la configuración y operación del protocolo EGP.
- **transmission**: no se definen actualmente objetos para este grupo, pero otros MIBs están definidos utilizado este subárbol.
- **snmp**: mide el rendimiento de la implementación SNMP en la entidad gestionada y realiza el seguimiento de cosas como el número de paquetes SNMP enviados y recibidos.

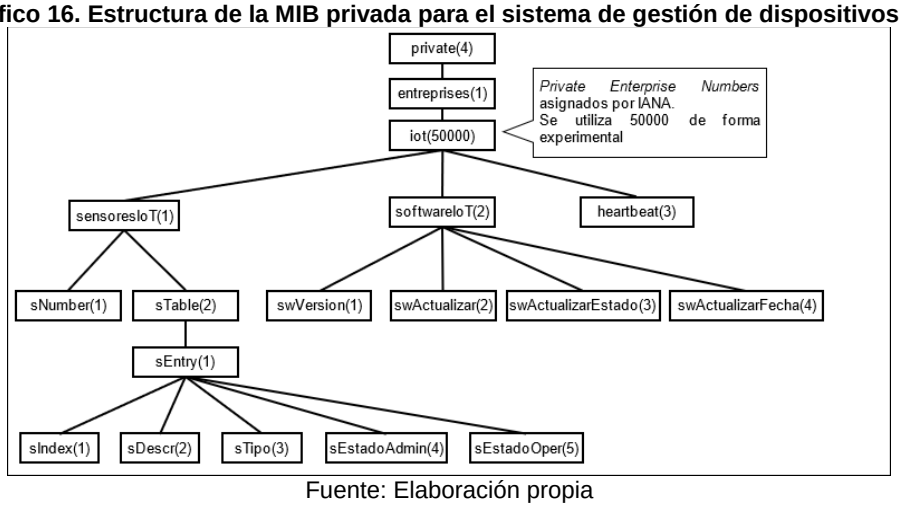
Para el sistema de gestión de IoT, no son aplicables las MIBs de los grupos at (por obsolescencia), egp (por no implementación de este protocolo en los dispositivos IoT) y transmission (pues no tiene objetos actualmente).

Adicionalmente a lo anterior, dada las características de los dispositivos IoT y de las aplicaciones que soportan, basadas principalmente en el uso de sensores, resulta necesario la incorporación de una MIB de tipo privado, con información específica de dichos dispositivos, que no se encuentra incluida en las MIBs de tipo estándar. La función de esta MIB es entregar información relacionada con el estado de diversos sensores que posee el dispositivo IoT.

Por otra parte, una de las características del sistema de gestión para redes IoT es la actualización del software de los dispositivos IoT. Para realizar esta función a través de SNMP se hace uso de una MIB de tipo privado, mediante la cual se le indica al dispositivo IoT la versión de software que debe descargar del servidor de archivos, pues como se había indicado anteriormente, SNMP no soporta la transferencia de archivos.

Con base en lo anterior, la MIB privada para IoT tiene la estructura que se observa en el Gráfico 16.

Gráfico 16. Estructura de la MIB privada para el sistema de gestión de dispositivos IoT



Fuente: Elaboración propia

En la Tabla 2 se muestra la descripción de los objetos del subárbol de sensores IoT, el cual está basado en el grupo interfaces de la MIB-II.

Tabla 2. Objetos de subárbol de sensores

OBJETO	DESCRIPCIÓN	SYNTAXIS	PERMISO	OID
sNumber	Número de sensores del dispositivo IoT	INTEGER	read-only	1.3.6.1.4.1.50000.1.1
sTable	Lista de sensores presentes en el dispositivo	SEQUENCE of sEntry	not-accessible	1.3.6.1.4.1.50000.1.2
sEntry	Entrada que contiene información de un sensor	sEntry	not-accessible	1.3.6.1.4.1.50000.1.2.1
sIndex	Valor único para cada sensor	INTEGER	read-only	1.3.6.1.4.1.50000.1.2.1.1
sDescr	Cadena de texto que contiene información del sensor	DisplayString	read-only	1.3.6.1.4.1.50000.1.2.1.2
sTipo	Tipo de sensor	INTEGER { Lista de tipos de sensores }	read-only	1.3.6.1.4.1.50000.1.2.1.3
sEstadoAdmin	Estado configurado de operación del sensor	INTEGER { up(1), down(2) }	read-write	1.3.6.1.4.1.50000.1.2.1.4
sEstadoOper	Estado actual de operación del sensor	INTEGER { up(1), down(2) }	read-only	1.3.6.1.4.1.50000.1.2.1.5

Fuente: Elaboración propia

Así mismo, en la Tabla 3 se muestra la descripción de los objetos del subárbol de software IoT, utilizada para la actualización de los dispositivos IoT, y que está basado en la MIB de Cisco OLD-CISCO-FLASH-MIB.

Tabla 3. Objetos del subárbol de software IoT

OBJETO	DESCRIPCIÓN	SYNTAXIS	PERMISO	OID
swVersion	Versión actual de software del dispositivo IoT	DisplayString	read-only	1.3.6.1.4.1.50000.2.1
swActualizar	Nombre de la versión de software a descargar del servidor sftp	DisplayString	write-only	1.3.6.1.4.1.50000.2.2
swActualizarEstado	Estado de la actualización actual o de la última	INTEGER { enProgreso(1), exitoso(2), fallo(3), noAfterPowerOn(4) }	read-only	1.3.6.1.4.1.50000.2.3
swActualizarFecha	Fecha de la última actualización	DisplayString	read-only	1.3.6.1.4.1.50000.2.4

Fuente: Elaboración propia

Por último, se incluyen otros objetos de información, relacionados con otras funciones:

Tabla 4. Otros objetos de información

OBJETO	DESCRIPCIÓN	SYNTAXIS	PERMISO	OID
heartbeat	Valor en segundos de cada cuanto el dispositivo IoT envía un heartbeat al Gateway IoT	INTEGER	read-only	1.3.6.1.4.1.50000.3.1

Fuente: Elaboración propia

Así mismo, para la gestión de alertas de fallas en los dispositivos IoT, se incluyen los siguientes traps:

Tabla 5. Traps

OBJETO	DESCRIPCIÓN	OID
heartbeat	Indica que el agente está disponible. El tiempo de envío de este trap está configurado en el OID heartbeat (1.3.6.1.4.1.50000.3.1)	1.3.6.1.6.3.1.1.5.50000.1
sensorDownAdmin	Indica que el agente ha detectado que el sensor ha cambiado su estado administrativo a DESACTIVADO. Mediante el valor de este trap se indica el sensor afectado.	1.3.6.1.6.3.1.1.5.50000.2
sensorUpAdmin	Indica que el agente ha detectado que el sensor ha cambiado su estado administrativo a ACTIVADO. Mediante el valor de este se trap indica el sensor afectado.	1.3.6.1.6.3.1.1.5.50000.3
sensorDownOper	Indica que el agente ha detectado que el sensor ha cambiado su estado operativo a DESACTIVADO. Mediante el valor de este se trap indica el sensor afectado.	1.3.6.1.6.3.1.1.5.50000.4
sensorUpOper	Indica que el agente ha detectado que el sensor ha cambiado su estado operativo a ACTIVADO. Mediante el valor de este trap se indica el sensor afectado.	1.3.6.1.6.3.1.1.5.50000.5
actualizacion	Indica que el dispositivo IoT ha terminado la actualización del software. Mediante el valor de este trap se indica el resultado de este proceso, de acuerdo con los indicados en el OID swActualizarEstado (1.3.6.1.4.1.50000.2.3), el cual puede ser: exitoso(2) o fallo(3). El resultado también puede ser consultado a través del mencionado OID.	1.3.6.1.6.3.1.1.5.50000.6

Fuente: Elaboración propia

3.1.3 Comunicación entre la aplicación de gestión y el Gateway IoT

La comunicación entre el Gateway IoT y la aplicación de gestión debe permitir no solo el intercambio de datos entre los dos nodos, sino que también el intercambio de archivos, con el fin de enviar a través de este medio el software a ser actualizado en los dispositivos IoT.

En materia de IoT, existen múltiples protocolos que permiten realizar el intercambio de datos entre los dispositivos y la nube, entre los cuales están:

- MQTT (Message Queuing Telemetry Transport)[17]: es un protocolo utilizado para la comunicación machine-to-machine (M2M), especialmente diseñado para escenarios donde el ancho de banda y los recursos (CPU, RAM) son limitados.

- SOAP (Simple Object Access Protocol)[18]: es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML, lo cual en principio permite intercambiar cualquier tipo de información. SOAP funciona por lo general haciendo uso de HTTP que es lo más común cuando se invoca un Web Services, sin embargo, no está limitado a este protocolo, si no que puede ser enviado por FTP, POP3, TCP, Colas de mensajería (JMS, MQ, etc). Permite establecer cifrado a nivel de los mensajes.

- REST (REpresentational State Transfer)[19]: describe un conjunto de principios de la arquitectura por el cual los datos se pueden transmitir a través de una interfaz estandarizada (como HTTP). Permite transmitir prácticamente cualquier tipo de datos como XML, JSON, Binarios (imágenes, documentos), Text, etc. REST hereda las medidas de seguridad del transporte subyacente, en este caso HTTPS.

[17] Estándar disponible en: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html
[18] Estándar disponible en: https://www.w3.org/TR/soap/
[19] https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf

Un aspecto importante que debe considerarse en la comunicación es que se permita el intercambio de archivos, por el que protocolo seleccionado debe permitir realizar esta tarea.

Cuando es importante considerar el atributo relacionado con el bajo consumo de recursos, en aplicaciones donde la comunicación se realiza directamente en los dispositivos IoT y la nube, los protocolos con bajo consumo de recursos son los ideales. Sin embargo, para aplicaciones basadas en Edge Computing, el consumo de recursos recae sobre los Gateway IoT, los cuales generalmente no se ven afectados por esto.

Por otra parte, teniendo en cuenta que la información que se va a intercambiar entre el Gateway IoT y la aplicación de gestión es bastante sensible, ya que no solo involucra el intercambio de datos sino en envío de acciones, la característica más relevante que debe tenerse en cuenta es la relacionada con la seguridad de la comunicación. En este aspecto, SOAP ofrece mejores prestaciones, a través del uso de Web Service Security (WS-Security)[20], que es un conjunto de mejoras a la mensajería SOAP que permite proteger el contenido de un mensaje de una modificación no autorizada y no detectada (integridad del mensaje) y proteger el contenido de un mensaje de una divulgación no autorizada (confidencialidad del mensaje). Con lo anterior, se asegura una seguridad de extremo a extremo, adicional a las medidas de seguridad características de HTTPS.

Con base en lo anterior, para la implementación de la comunicación entre la aplicación de gestión y el Gateway IoT se utiliza SOAP, principalmente por las características de seguridad que ofrece. En la Tabla 6 se muestra una comparación de las características antes mencionadas.

Tabla 6. Comparación de opciones de comunicación entre el Gateway IoT y la aplicación de gestión

Atributo	MQTT	SOAP	REST
Permite intercambio de datos	Si	Si	Si
Permite intercambio de archivos	No	Si	Si
Consumo de recursos	Bajo	Alto	Medio
Medidas de seguridad	Medio (SSL/TLS)	Alto (WS-Security y SSL/TLS)	Medio (SSL/TLS)

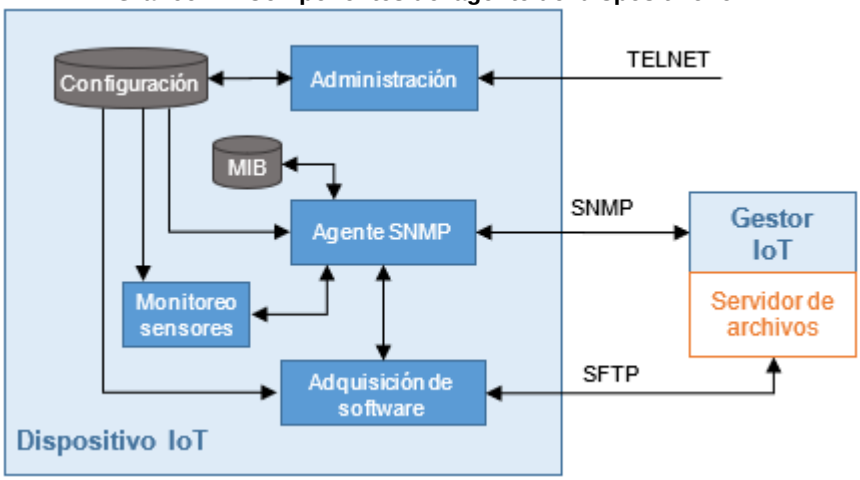
Fuente: Elaboración propia

3.2 DISEÑO DEL AGENTE DEL DISPOSITIVO IOT

Teniendo en cuenta las funciones de gestión en el dispositivo IoT, el diseño desarrollado para el agente está conformado por 4 módulos, tal y como puede observarse en el Gráfico 17.

[20] https://www.ibm.com/support/knowledgecenter/es/SSMKHH_10.0.0/com.ibm.etools.mft.doc/ac55970_.htm

Gráfico 17. Componentes del agente del dispositivo IoT



Fuente: Elaboración propia

A continuación se realiza la descripción de cada uno de los módulos:

3.2.1 Agente SNMP

En este módulo implementa las funcionalidades del protocolo SNMP, desde el punto de vista del agente, que como se indicó previamente, corresponde a la versión 2c, teniendo en cuenta que posteriormente puede ser implementada la versión 3.

Teniendo en cuenta lo anterior, este módulo se encarga de realizar las siguientes funciones:

- Recepción de mensajes de SNMP, relacionado con la lectura (GetRequest, GetNextRequest) y escritura (SetRequest) de valores de los objetos de información.
- Envío de mensajes SNMP hacia el Gateway IoT, para dar respuesta a mensajes enviados por el gestor (GetResponse) y para reportar ciertas condiciones o cambios en el dispositivo IoT (Trap). En cuanto a este último tipo de mensajes, el dispositivo IoT tendrá en cuenta los tipos de Traps indicados en la Tabla 5.
- Lectura/escritura de las MIBs generales, específicamente aquellas relacionadas con las de tipo *system*.
- Lectura/escritura de las MIBs específicas diseñadas para los dispositivos IoT, las cuales están indicadas en el Gráfico 16.
- Interacción con módulo de adquisición de software, para actualización del dispositivo IoT. A través de mensajes SNMP enviados por el Gateway SNMP, el módulo Agente SNMP debe indicarle al módulo “Adquisición de software” que realice la descarga del archivo ara ejecutar la actualización. En esta función se hacen uso de los objeto de información indicados en la Tabla 3.

Adicionalmente, este módulo tiene asociado una base de datos de MIB, en la cual se almacena la información del protocolo SNMP.

3.2.2 Monitoreo de sensores

Este módulo se encarga de monitorear el estado de sensores, es decir, si están activados o desactivados, ya sea como consecuencia de procesos de administración o por operación.

Teniendo en cuenta lo anterior, este módulo se encarga de realizar las siguientes funciones:

- Lectura periódica del estado administrativo de los sensores. Es importante tener en cuenta que el estado administrativo de los sensores se puede cambiar a través objeto de información *sEstadoAdmin.*
- Lectura periódica del estado operativo de los sensores. Es importante tener en cuenta que el objeto de información *sEstadoOper* es de tipo de solo lectura, teniendo en cuenta que este estado depende precisamente de la operatividad del sensor.
- Detectar cambios en el estado (administrativo y operativo) de los sensores y enviar Traps al Gateway IoT, a través del módulo “Agente SNMP”. Para esto, hace uso de los objetos de información indicados en la Tabla 5

3.2.3 Adquisición de software

Este módulo se encarga de realizar la actualización del software del dispotivo IoT, asociado a la aplicación de IoT que se está ejecutando. Es importante tener en cuenta que esto no incluye actualización del firmware de dispositivo IoT o de su sistema operativo.

Teniendo en cuenta lo anterior, este módulo se encarga de realizar las siguientes funciones:

- Descarga de software desde el Gateway IoT, mediante la instrucción enviada por el módulo "Agente SNMP".
- Realizar la actualización del dispositivo IoT, posterior a la descarga del software.
- Actualizar los objetos de información relacionados con la actualización del software, indicados en la Tabla 3:
  - swVersion: ajusta la versión del software del dispositivo IoT, en caso de haberse realizado correctamente la actualización
  - swActualizarEstado: Cambio el estado de la actualización, en la medida en la que vaya avanzando dicho proceso:
    - enProgreso: indica que actualmente se está realizado una actualización, lo cual incluye la descarga del software del dispositivo IoT
    - exitoso: indica que el último proceso de actualización ha sido exitoso
    - fallo: indica que el último proceso de actualización ha fallado
    - noAfterPowerOn: indica que no se ha ejecuta ningún proceso de actualización de software después del último ciclo de encendido del dispositivo ioT
  - swActualizarFecha: ajusta la fecha de actualización, en caso de haberse realizado correctamente la actualización.

3.2.4 Administración

Mediante este módulo se realiza la configuración vía remota del dispositivo IoT, a través del acceso y modificación a la base de datos de configuración, la cual contiene la siguiente información:

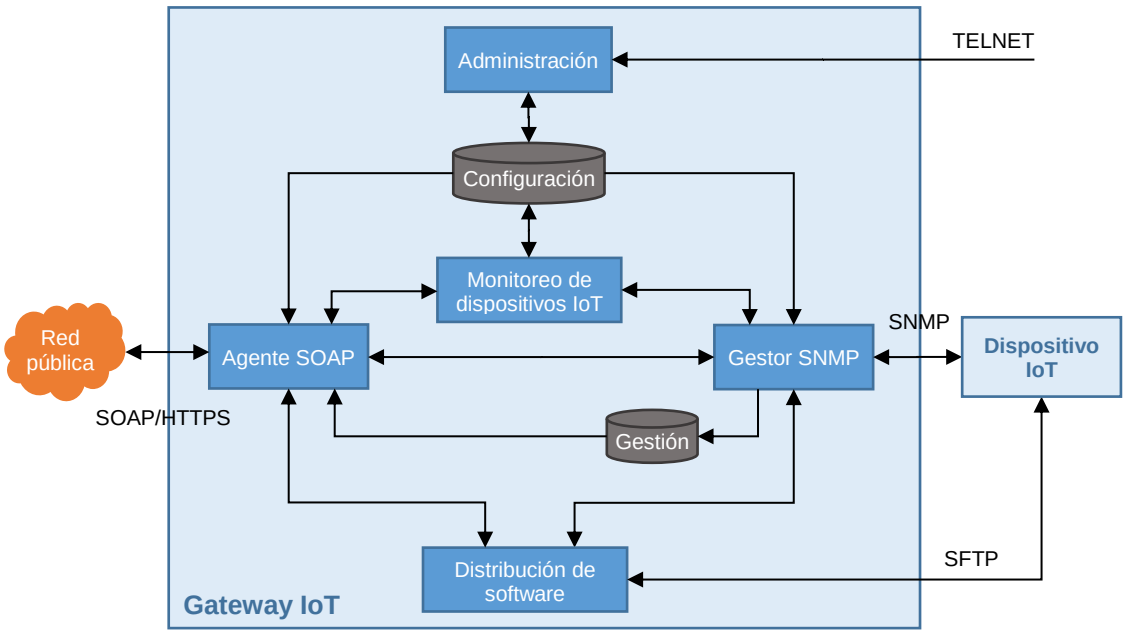
- Parámetros de conexión al módulo de administración
  - Usuario, contraseña y puerto de acceso
- Parámetros del protocolo SNMP:
  - Puerto del protocolo SNMP
  - Protocolo de capa de transporte: TCP o UDP
  - Comunidad SNMP
  - Dirección IP, puerto y protocolo de capa de transporte del gestor, para efectos de envío de los traps. Pueden ser más de un gestor.
- Parámetros del servidor remoto de actualización de software:
  - Dirección IP y puerto del servidor de actualización de software
  - Usuario y contraseña de conexión al servidor de actualización de software

- Carpeta a la que va a realizar la conexión para descargar el software
- Parámetros de los sensores:
  - Tipo de sensor
  - Nombre del sensor
  - Tiempo de monitoreo del sensor, para efector de verificar su estado

3.3 DISEÑO DEL GESTOR DEL GATEWAY IOT

Teniendo en cuenta las funciones de gestión en el Gateway IoT, el gestor está conformado por 5 módulos, tal y como puede observarse en el Gráfico 18.

Gráfico 18. Componentes del gestor del Gateway IoT



Fuente: Elaboración propia

A continuación se realiza la descripción de cada uno de los módulos:

3.3.1 Gestor SNMP

En este módulo implementa las funcionalidades del protocolo SNMP, desde el punto de vista del gestor, que como se indicó previamente, corresponde a la versión 2c, teniendo en cuenta que posteriormente puede ser implementada la versión 3.

Teniendo en cuenta lo anterior, este módulo se encarga de realizar las siguientes funciones:

- Envío de mensajes SNMP hacia los dispositivos IoT, para la consulta y modificación de parámetros de los mismo, dentro de los cuales están los siguientes tipo:
  - GetRequest: petición de valores específicos de la MIB.
  - GetNextRequest: petición del objeto siguiente a uno dado de la MIB.
  - GetBulkRequest: petición de múltiples valores.
  - SetRequest: permite asignar un valor a una variable.

- Recepción de mensajes de respuesta y traps del agente de los dispositivos IoT, relacionados con las respuestas a las peticiones realizados y a los traps. En cuanto a este último tipo de mensajes, el Gateway IoT tendrá en cuenta los tipos de Traps indicados en la Tabla 5.
- Almacenamiento de la información obtenida de los dispositivos IoT en una base de datos de gestión.
- Recepción/envío mensajes de la aplicación de gestión de la nube, con su correspondiente acción hacia los dispositivos IoT:
  - Cambio de estado administrativo de un sensor del dispositivo IoT
  - Actualización del software de los dispositivos IoT
  - Reinicio de un dispositivo IoT, a través del objeto de información *restart*, indicado en la Tabla 4.

Adicionalmente, este módulo tiene asociado una base de datos de gestión, en la cual se almacena la información recibida de los dispositivos IoT.

3.3.2 Monitoreo de dispositivos IoT

Este módulo se encarga de monitorear el estado de los dispositivos IoT, es decir, si están disponibles o no. Teniendo en cuenta lo anterior, este módulo se encarga de realizar las siguientes funciones:

- Determinar si el agente SNMP de los dispositivos IoT están disponibles, es decir, si responden a mensajes de heartbeat, y a través del establecimiento de temporizadores de revisión de estado de conexión.
- Detección de cambios en el estado de dispositivos IoT, cuando se vencen los temporizadores. Para esto, hace uso del Trap *heartbeat* que recibe el módulo “Gestor SNMP”, y que se describe en la Tabla 5
- Envío mensaje a la aplicación de gestión en la nube, a través del módulo “Agente SOAP”, cuando se detectan cambios en la disponibilidad de los dispositivos IoT

3.3.3 Agente SOAP

Este módulo se encarga de realizar la comunicación entre el Gateway IoT y la aplicación de gestión, a través del protocolo SOAP, como se indicó previamente en el presente documento.

Teniendo en cuenta lo anterior, este módulo se encarga de realizar las siguientes funciones:

- Envío y recepción de información de los dispositivos IoT a la aplicación de gestión, recibida a través del módulo “Gestor SNMP”.
- Traducción de las instrucciones recibidas de la aplicación de gestión, para remitirlas al módulo “Gestor SNMP”, quien se encargará de enviarlas a los dispositivos IoT.
- Interacción con módulo “Distribución de software”, para actualización de los dispositivos IoT. A través de mensajes enviados por la aplicación de gestión, el módulo “Agente SOAP” debe enviarle al módulo “Distribución de software” el software de actualización.

3.3.4 Distribución de software

Este módulo se encarga de realizar la distribución del software de actualización a los dispositivos IoT.

Teniendo en cuenta lo anterior, este módulo se encarga de realizar las siguientes funciones:

- Interacción con módulo “Agente SOAP”, quien le enviará el software de actualización
- Solicitar la actualización de software de los dispositivos IoT a través del módulo “Gestor SNMP”.

- Informar a la aplicación de gestión, a través del módulo “Agente SOAP”, sobre el proceso de actualización de software de los dispositivos IoT.

3.3.5 Administración

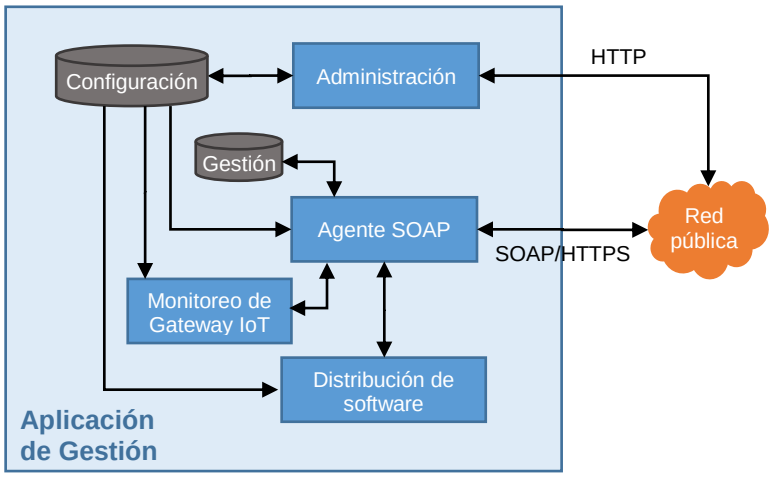
Mediante este módulo se realiza la configuración vía remota del Gateway IoT, a través del acceso y modificación a la base de datos de configuración, la cual contiene la siguiente información:

- Parámetros de conexión al módulo de administración
  - Usuario, contraseña y puerto de acceso
- Parámetros del protocolo SNMP
  - Puerto del protocolo SNMP
  - Protocolo de capa de transporte: TCP o UDP
  - Comunidad SNMP
  - Puerto de Traps
- Parámetros de los dispositivos IoT
  - Dirección IP, puerto SNMP, comunidad y protocolo de capa de transporte del dispositivo IoT.
- Parámetros de la aplicación de gestión
  - Dirección IP, puerto, usuario y contraseña para la comunicación
- Parámetros del servidor remoto de actualización de software

3.4 DISEÑO DE LA APLICACIÓN DE GESTIÓN DE LA PLATAFORMA DE COMPUTACIÓN EN LA NUBE

Teniendo en cuenta las funciones de gestión en la aplicación de gestión, esta está conformada por 4 módulos, tal y como puede observarse en el Gráfico 19.

Gráfico 19. Componentes de la aplicación de gestión en la nube



Fuente: Elaboración propia

3.4.1 Agente SOAP

Este módulo se encarga de realizar la comunicación entre la aplicación de gestión y el Gateway IoT, a través del protocolo SOAP, como se indicó previamente en el presente documento.

Teniendo en cuenta lo anterior, este módulo se encarga de realizar las siguientes funciones:

- Envío y recepción de información de los dispositivos IoT.
- Cambiar parámetros de configuración de los dispositivos IoT
- Recepción de alarmas enviadas por los dispositivos IoT a través del Gateway IoT
- Interacción con módulo de distribución de software, para actualización de los dispositivos IoT.

Adicionalmente, este módulo tiene asociado una base de datos de gestión, en la cual se almacena la información recibida de los dispositivos IoT, recibida de los Gateway IoT.

3.4.2 Monitoreo de Gateway IoT

Este módulo se encarga de monitorear el estado de los Gateway IoT. Teniendo en cuenta lo anterior, este módulo se encarga de realizar las siguientes funciones:

- Determinar el estado de disponibilidad de los Gateway IoT, a través del establecimiento de temporizadores de revisión de estado de conexión.
- Detección de cambios en el estado de los Gateway IoT, cuando se vencen los temporizadores.
- Actualización de la información de conexión de los Gateway IoT en la base de datos de gestión, cuando se detectan cambios en la disponibilidad de los mismos.

3.4.3 Distribución de software

Este módulo se encarga de realizar la distribución del software de actualización a los dispositivos IoT. Para lo anterior, realiza la interacción con módulo “Agente SOAP”, para el envío del software de actualización a los Gateway IoT

3.4.4 Administración

Mediante este módulo se realiza la configuración vía remota de parámetros de la aplicación de gestión de la nube, la cual es almacenada en una base de datos de configuración. Así mismo, a través de este módulo se realiza la visualización y modificación de la información de los dispositivos IoT enviada por los Gateway IoT, a través de una interfaz Web.

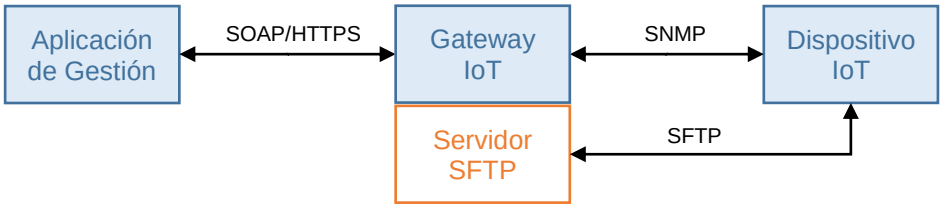
3.5 CONSIDERACIONES ADICIONALES PARA EL DISEÑO

Uno de los aspectos que debe tenerse en cuenta, y que ya fue mencionado anteriormente, es que a través de SNMP no es posible realizar la función de actualización de software de los dispositivos IoT, y que, por lo tanto, se implementa una solución de transferencia de archivos.

Una de las soluciones más sencillas es el FTP, que es un método popular de transferencias de archivos, pero no tiene en cuenta medidas de seguridad. Por lo anterior, surgió SFTP, un protocolo completamente diferente basado en el protocolo SSH (Secure Shell).

Con base en lo anterior, la solución para la gestión de dispositivos IoT a través de SNMP, requiere de la existencia de un servidor SFTP al cual se conectan estos para realizar la descarga del archivo de actualización del software.

Gráfico 20. Solución para actualización del software de los dispositivos IoT



Fuente: Elaboración propia

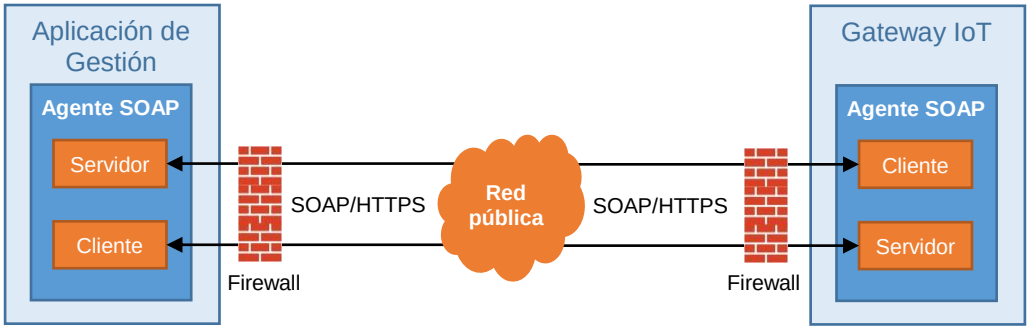
Por otra parte, la implementación del agente SOAP en el Gateway IoT requiere de 2 puertos de comunicaciones:

- Un puerto en el que el agente actúa como cliente, mediante el cual el Gateway IoT envía información a la aplicación de gestión.
- Un puerto en el que el agente actúa como servidor, mediante el cual el Gateway IoT ejecuta acciones iniciadas por la aplicación de gestión.

Estas mismas consideraciones aplican para el agente SOAP de la aplicación de gestión, es decir, que también requiere de 2 puertos de comunicaciones.

Lo anterior requiere de la apertura de estos puertos de cara a la red pública de Internet, y su redirección hacia el Gateway IoT y la aplicación de gestión.

Gráfico 21. Implementación de la comunicación entre la aplicación de gestión y el Dispositivo IoT



Fuente: Elaboración propia

3.6 DISEÑO DE PROCESOS

3.6.1 Inicialización de dispositivos

3.6.1.1 Inicio del dispositivo IoT

En el proceso de encendido del dispositivo IoT, se realizan las siguientes acciones:

1. Inicio de los módulos del dispositivo IoT:
   a. Agente SNMP: En estado de espera de mensajes SNMP por parte del Gateway IoT, en el puerto configurado, y de acciones por parte de los demás módulos.
   b. Monitoreo de sensores: Lectura de los estados administrativo y operativo de cada uno de los sensores incluidos en la base de datos de configuración, de acuerdo con el valor de tiempo configurado.
   c. Administración: es espera de conexiones Telnet, en el puerto configurado.

d. Adquisición de software: en espera de orden para descarga de archivo de software, desde el servidor SFTP configurado.
2. Envío de un Trap de tipo *coldstart* hacia el Gateway IoT.
3. Envío de Traps de tipo *heartbeat* hacia el Gateway IoT, de acuerdo con el valor de tiempo configurado.

3.6.1.2 Inicio del Gateway IoT

En el proceso de encendido del gateway IoT, se realizan las siguientes acciones:

1. Inicio de los módulos del gateway IoT:
   a. Gestor SNMP: En estado de espera de mensajes SNMP por parte del dispositivo IoT, en el puerto configurado, y de acciones por parte de los demás módulos.
   b. Agente SOAP: En estado de espera de mensajes SOAP por parte de la aplicación de gestión, y de acciones por parte de los demás módulos.
   c. Monitoreo de dispositivos IoT: En estado de espera de detección de dispositivos IoT configurados. Inicialmente, todos los dispositivos IoT configurados se clasifican como "no disponibles".
   d. Administración: es espera de conexiones Telnet, en el puerto configurado.
   e. Distribución de software: en espera de nuevo archivo de software, para solicitar actualización a los dispositivos IoT configurados.
2. Envío de un mensaje SOAP hacia la aplicación de gestión con la información de los dispositivos IoT configurados.
3. Envío de mensajes SOAP de tipo *heartbeat* hacia la aplicación de gestión, de acuerdo con el valor de tiempo configurado.

3.6.1.3 Inicio de la aplicación de gestión

En el proceso de encendido de la aplicación de gestión, se realizan las siguientes acciones:

1. Inicio de los módulos del gateway IoT:
   a. Agente SOAP: En estado de espera de mensajes SOAP por parte del Gateway IoT, y de acciones por parte de los demás módulos.
   b. Monitoreo de gateway IoT: En estado de espera de detección de gateway IoT configurados.
   c. Administración: es espera de conexiones Web, en el puerto configurado.
   d. Distribución de software: en espera de nuevo archivo de software, para enviarlo a los dispositivos IoT configurados.

3.6.2 Detección de un dispositivo IoT parte del Gateway IoT

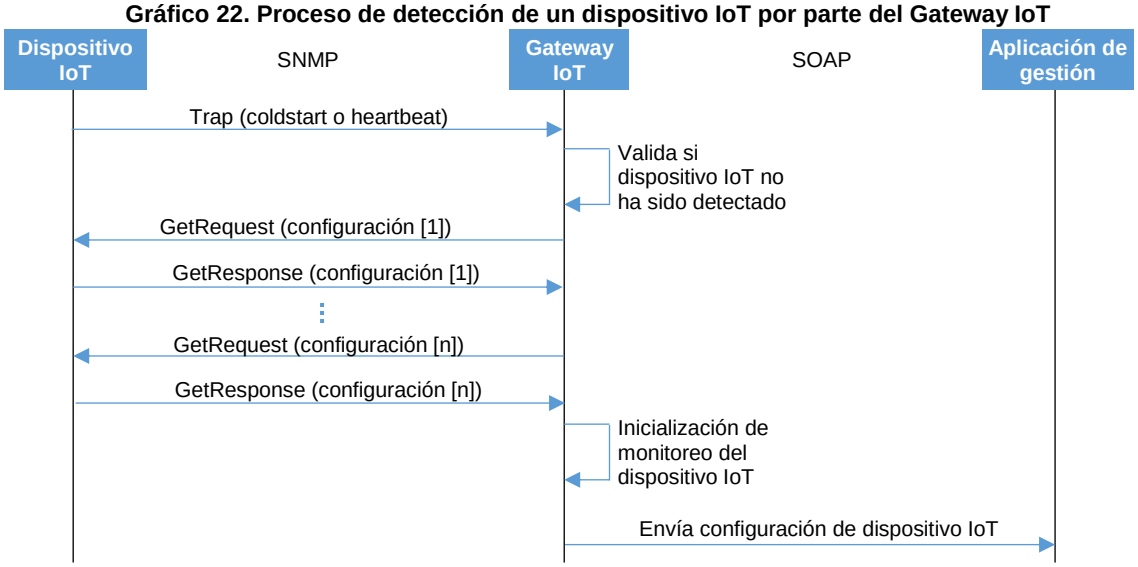
La detección de los dispositivos IoT puede ser realizada de 2 formas por parte del Gateway IoT: mediante el trap de tipo *coldstart* o a través de los traps de tipo *heartbeat* que envía periódicamente el dispositivo IoT

Una vez es detectado alguno de estos traps, el Gateway identifica si el dispositivo IoT ya ha sido detectado, consultando su base de datos de configuración. En caso de no haber sido detectado, realiza las siguientes acciones:

1. El Gateway IoT solicita la configuración al dispositivo IoT sobre:
   a. Los OID del subárbol system.
   b. Los OID del subárbol de sensores IoT
   c. La versión de software. (Si la versión de software es inferior a la requerida, solicita al dispositivo IoT su actualización. Este proceso es explicado más adelante).
   d. El OID de heartbeat

2. El Gateway IoT actualiza su base de datos de configuración, cambiando el estado del dispositivo IoT a “disponible”.
3. El Gateway IoT inicializa el monitoreo del dispositivo IoT, a través del establecimiento de un temporizador basado en el valor del OID de heartbeat.
4. El Gateway IoT actualiza la información del dispositivo IoT hacia la aplicación de gestión.
5. La aplicación de gestión actualiza su base de datos de configuración.

Gráfico 22. Proceso de detección de un dispositivo IoT por parte del Gateway IoT



Es importante tener en cuenta que cuando el Gateway IoT recibe un trap de tipo *coldstart*, siempre realiza las acciones indicadas anteriormente. En el caso del trap de tipo heartbeat, únicamente las realiza si previamente ha catalogado el dispositivo IoT como no disponible, proceso que se explica más adelante.

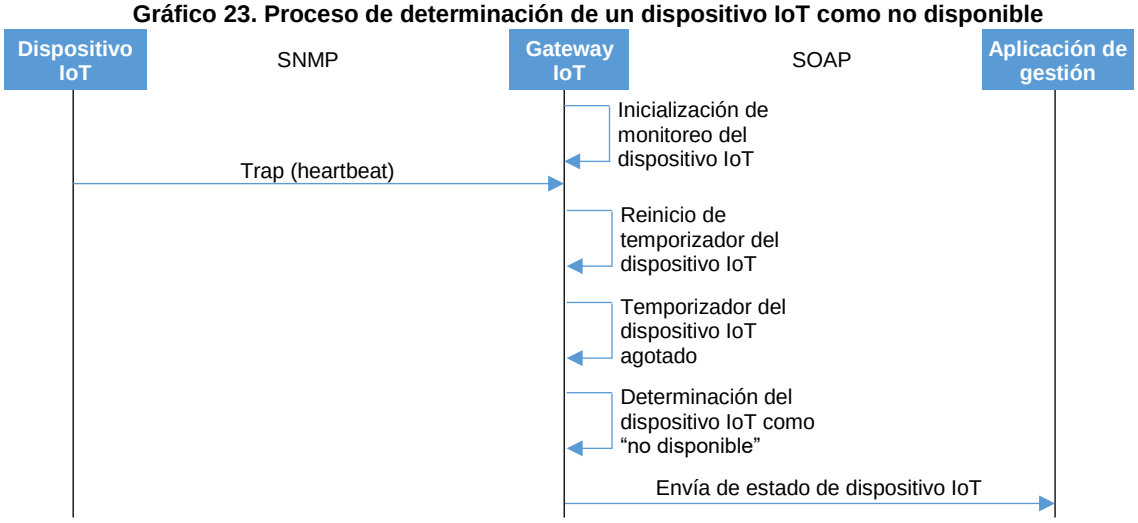
3.6.3 Determinación de un dispositivo IoT como “no disponible”

Una vez se ha iniciado el monitoreo de un dispositivo IoT, el Gateway IoT inicializa un temporizador basado en el valor del OID heartbeat previamente consultado a dicho dispositivo, así como un contador de heartbeat que se inicializa en un valor máximo.

Cada vez que se agota el temporizador, el contador de hearbeat se decrementa en 1. Sin embargo, cada vez que se recibe un trap de tipo *heartbeat*, este contador se devuelve a su valor máximo.

Cuando contador de hearbeat llega a “0”, se considera al dispositivo IoT como “no disponible”, y el Gateway IoT envía un mensaje SOAP a la aplicación de gestión para informar sobre el cambio de estado. La aplicación de gestión recibe esta información y actualiza su base de datos de configuración.

Gráfico 23. Proceso de determinación de un dispositivo IoT como no disponible



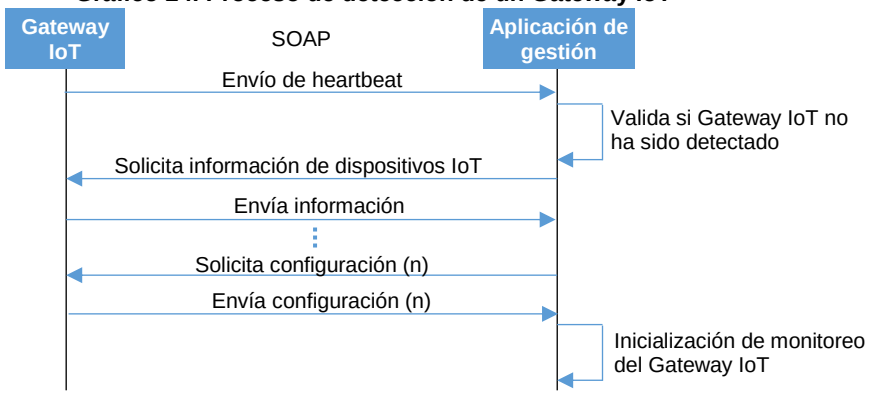
3.6.4 Detección de un Gateway IoT por parte de la aplicación de gestión

La detección de los Gateway IoT por pate de la aplicación de gestión se realiza mediante los mensajes SOAP tipo *heartbeat* que envía periódicamente el Gateway IoT.

Una vez recibe un mensaje SOAP de tipo heartbeat, la aplicación de gestión identifica si el gateway IoT ya ha sido detectado, consultando su base de datos de configuración. En caso de no haber sido detectado, realiza las siguientes acciones:

1. Solicita la información de los dispositivos IoT conectado al Gateway IoT:
   a. Los OID del subárbol system.
   b. Los OID del subárbol de sensores IoT
   c. La versión de software
2. Actualiza su base de datos de configuración, cambiando el estado del Gateway IoT a "disponible".
3. Inicializa el monitoreo del Gateway IoT, a través del establecimiento de un temporizador.

Gráfico 24. Proceso de detección de un Gateway IoT



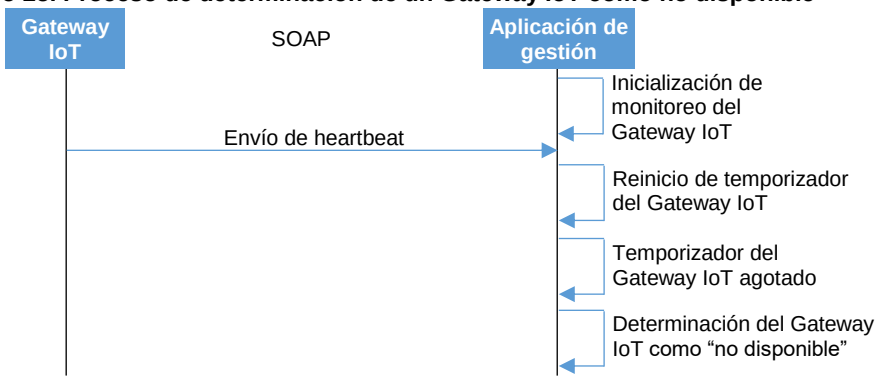
3.6.5 Determinación de un Gateway IoT como "no disponible"

Una vez se ha iniciado el monitoreo de un Gateway IoT, la aplicación de gestión inicializa un temporizador, así como un contador de heartbeat que se inicializa en un valor máximo.

Cada vez que se agota el temporizador, el contador de hearbeat se decrementa en 1. Sin embargo, cada vez que se recibe un mensaje SOAP de tipo *heartbeat*, este contador se devuelve a su valor máximo.

Cuando contador de hearbeat llega a “0”, se considera al Gateway IoT como “no disponible” y se actualiza la base de datos de configuración. Así mismo, todos los dispositivos IoT asociados al Gateway IoT son considerados como “no disponibles”.

Gráfico 25. Proceso de determinación de un Gateway IoT como no disponible



3.6.6 Detección de cambio en el estado de un sensor del dispositivo IoT

Una vez se ha iniciado el dispositivo IoT, este realiza la lectura de los estados administrativo y operativo de cada uno de los sensores incluidos en su base de datos de configuración, de acuerdo con el valor de tiempo configurado.

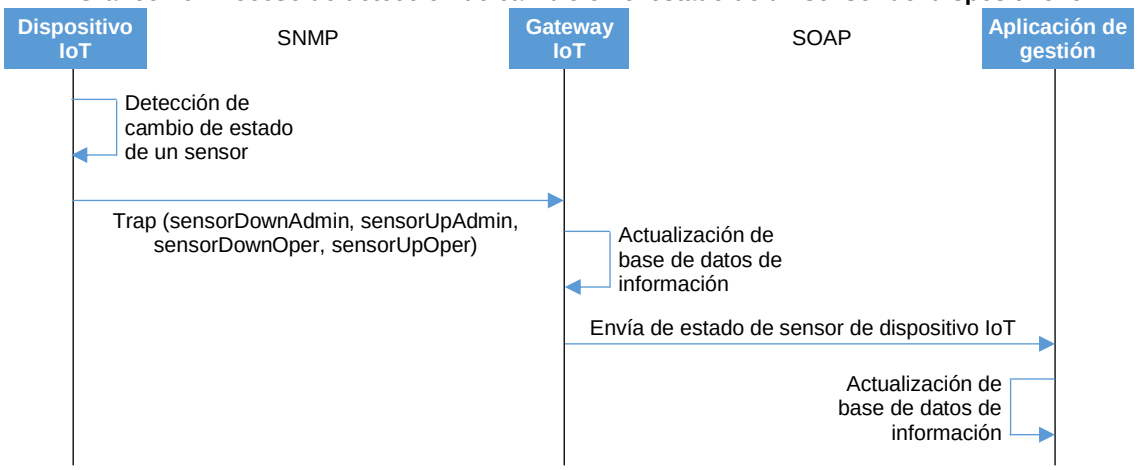
Si al leer los estados de un sensor se detecta un cambio en los mismos, el dispositivo IoT envía un trap al Gateway IoT para informar sobre esta situación. Los traps utilizados son los indicados en la Tabla 5

1. sensorDownAdmin: Indica que el sensor ha cambiado su estado administrativo a DESACTIVADO
2. sensorUpAdmin: Indica que el sensor ha cambiado su estado administrativo a ACTIVADO:
3. sensorDownOper: Indica que el sensor ha cambiado su estado operativo a DESACTIVADO.
4. sensorUpOper: Indica que el sensor ha cambiado su estado operativo a ACTIVADO

Una vez el Gateway IoT recibe alguno de estos traps, actualiza su base de datos de configuración, e informa a la aplicación de gestión, a través de un mensaje SOAP, sobre el cambio en el estado un sensor.

Posteriormente, la aplicación recibe el mensaje SOAP con esta información y actualiza su base de datos de configuración.

Gráfico 26. Proceso de detección de cambio en el estado de un sensor del dispositivo IoT

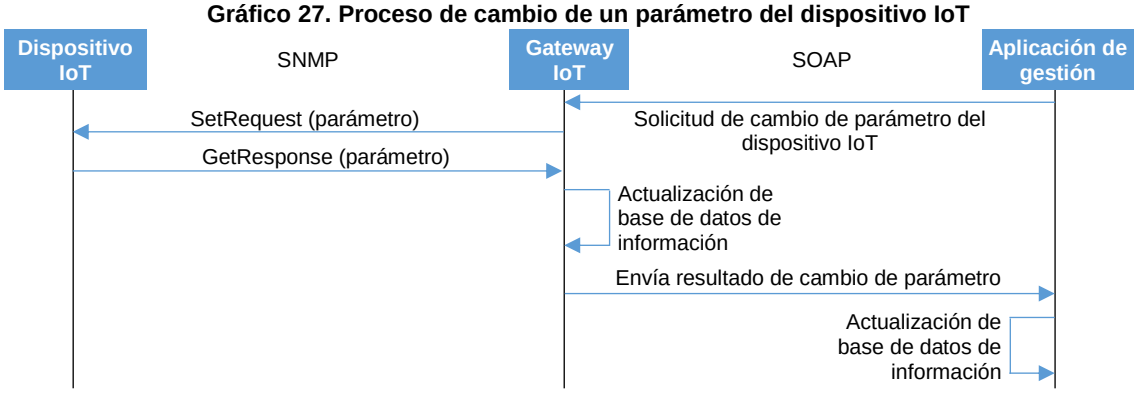


3.6.7 Cambio de un parámetro del dispositivo IoT

Para realizar el cambio de un parámetro de un dispositivo IoT, se sigue el siguiente procedimiento:

1. La aplicación de gestión envía un mensaje SOAP con la solicitud de cambio del valor de un parámetro al Gateway IoT
2. El Gateway IoT recibe el mensaje y envía la solicitud al dispositivo IoT a través de un mensaje SNMP *SetRequest*.
3. El dispositivo IoT realiza el cambio del parámetro e informe la respuesta al dispositivo IoT a través de un mensaje SNMP *GetResponse*.
4. El Gateway IoT recibe el mensaje SNMP, actualiza su base de datos de configuración y envía un mensaje SOAP a la aplicación de gestión con el resultado.
5. La aplicación de gestión recibe el mensaje SOAP y actualiza su base de datos de configuración.

Gráfico 27. Proceso de cambio de un parámetro del dispositivo IoT



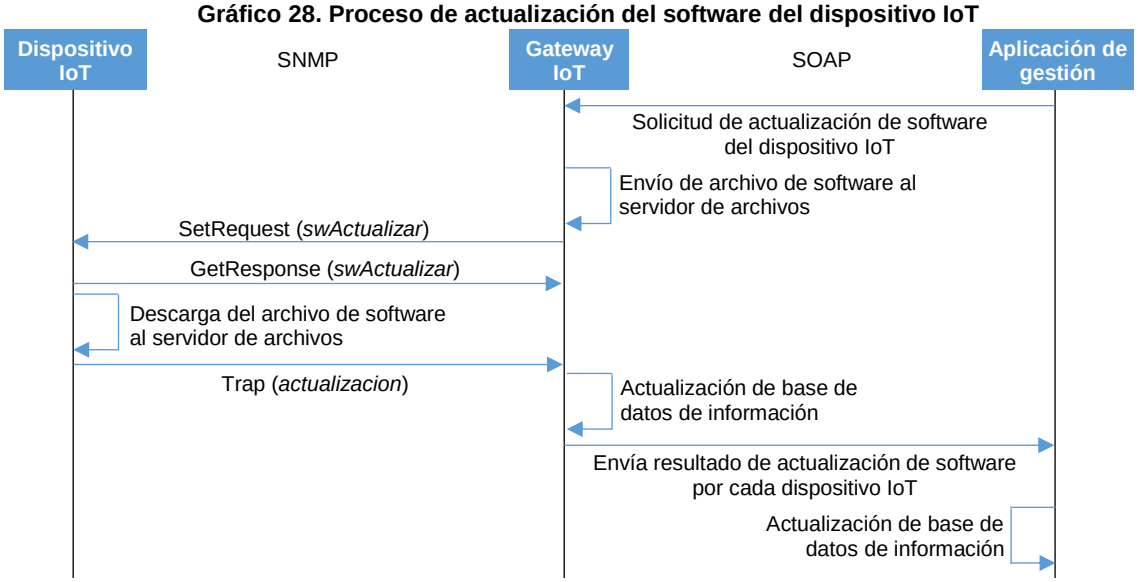
3.6.8 Actualización del software del dispositivo IoT

Para realizar el cambio de software de los dispositivos IoT, se sigue el siguiente procedimiento:

1. La aplicación de gestión envía un mensaje SOAP con la solicitud de actualización de software al Gateway IoT, el que va incluida el archivo.
2. El Gateway IoT recibe el mensaje, y envía el archivo al servidor SFTP, desde el cual los dispositivos IoT van a realizar la descarga del software.

3. El Gateway IoT envía la solicitud de actualización a los dispositivos IoT que está en estado “disponible”, a través de un mensaje SNMP utilizando el OID *swActualizar*, descrito en la Tabla 3.
4. El dispositivo IoT recibe el mensaje, y realiza la descarga del archivo indicado a través del servidor SFTP previamente configurado.
5. Una vez se completa la descarga, el dispositivo IoT realiza la actualización del software, e informa al Gateway IoT a través del trap *actualizacion* que el proceso ha terminado y su resultado.
6. Adicionalmente. el Gateway IoT puede verificar el resultado de la actualización, consultando los OID *swVersion*, *swActualizarEstado* y *swActualizarFecha* del dispositivo IoT.
7. El Gateway IoT, actualiza su base de datos de configuración y envía un mensaje SOAP a la aplicación de gestión con el resultado.
8. La aplicación de gestión recibe el mensaje SOAP y actualiza su base de datos de configuración.

Gráfico 28. Proceso de actualización del software del dispositivo IoT



Los dispositivos IoT que se encuentran en estado “no disponible” no son actualizados mediante el procedimiento descrito anteriormente. Su actualización se realiza cuando son detectados como “disponibles” por el Gateway IoT, y si su versión de software es inferior a la requerida, y para lo cual se realizan los pasos del 3 en adelante.

4 IMPLEMENTACIÓN DEL PROTOTIPO

En esta sección del documento se presentan los elementos de hardware y software utilizados en la implementación del prototipo del sistema de gestión de red

Es importante tener en cuenta que, por razones de tiempo y costo, el prototipo del sistema de gestión de red únicamente está en capacidad de realizar la función de gestión de fallos, la cual se encarga de detectar los fallos en los dispositivos IoT, lo cual incluye información de disponibilidad de los mismos y de sus sensores.

A pesar de que el sistema no contará con las demás funciones de un sistema de gestión de red, la implementación se realiza teniendo en cuenta que sea extensible para incluirlas a futuro.

4.1 LENGUAJE DE PROGRAMACIÓN

El primer elemento a tener en cuenta para la implementación del prototipo es la selección del lenguaje de desarrollo. Dentro de las opciones identificadas están el lenguaje C y java, y sus ventajas y desventajas, de caras a la implementación del prototipo, se muestran en la Tabla 7.

Tabla 7. Lenguajes de programación

LENGUAJE	VENTAJAS	DESVENTAJAS
C	• Bajo consumo de recursos	• Desconocimiento por parte del desarrollador • Curva de aprendizaje elevada
Java	• Conocimiento por parte del desarrollador • Curva de aprendizaje más baja	• Consumo de recursos más elevado, debido al uso de una máquina virtual de java (JVM)

Fuente: Elaboración propia

De acuerdo con lo anterior, el lenguaje adecuado para la implementación del prototipo es Java, debido al conocimiento del mismo por parte del desarrollador, a pesar de que consume más recursos de máquina que el lenguaje C. No obstante, para futuras implementaciones es posible considerar este último lenguaje.

4.2 DISPOSITIVO IOT

Para la implementación del dispositivo IoT, y teniendo en cuenta que el lenguaje de desarrollo es Java, se selección una plataforma de desarrollo que lo soporte, sin que esto implique que sea la única forma de implementarlo. En el mercado existe una multiplicidad de plataformas que soportan una máquina virtual de java. Sin embargo, por razones de costo y disponibilidad inmediata[21], fue seleccionado una Raspberry Pi Zero W, la cual tiene un procesador de un núcleo a 1GHz, 512 MB de memoria RAM y conectividad a través de Wifi 802.11n.

En cuanto a los sensores, igualmente por razones de costo y disponibilidad inmediata, se seleccionaron 2 tipos de sensores, que ofrecen diferentes tipos de acceso a las mediciones:

- HC-SR04: Este es un sensor que mide distancia mediante ultrasonido, a través del tiempo de envío y recepción de un pulso.
- DS18B20: es un sensor digital de temperatura que utiliza el protocolo 1-Wire para comunicarse, el cual necesita un solo pin de datos.

Para poder acceder a los sensores de manera controlada, se desarrolló un driver (módulo de kernel de Linux) para cada sensor, con el fin de que el agente SNMP pueda controlar el acceso a los mismos. A pesar de que el lenguaje de programación es Java, los drivers deben ser desarrollados en C. Los drivers de los sensores tienen los atributos que se observan en la Tabla 8.

Tabla 8. Atributos de los drivers de los sensores

ATRIBUTO	DESCRIPCIÓN
on	Activar el sensor
off	Desactivar el sensor
status	Muestra el estado del sensor (activado o desactivado)
status_oper	Muestra el estado operativo del sensor. Para poder obtener este estado, el driver realiza una medición. Si es exitosa devuelve “1”, y si no lo es devuelve “0”
measure (HC-SR04) w1_slave (DS18B20)	Indica al sensor que realicé una medición y devuelve el valor medido. Este atributo no es utilizado por el agente SNMP, pero fue diseñado para las aplicaciones que hagan uso del sensor. Es decir, las aplicaciones no deben acceder de manera directa al sensor, sino que deben hacerlo a través del driver

Fuente: Elaboración propia

[21] En el mercado colombiano se consigue por $100.000 a través de www.mercadolibre.com.co.

A partir de lo anterior, los módulos del dispositivo IoT se implementaron mediante el uso de hilos:

- Administración: A través de un hilo se crea el servidor Telnet, el cual a su vez lanza nuevos hilos por cada conexión establecida

- Agente SNMP: para este módulo se utilizó la librería SNMP4J[22], la cual es gratuita y de código abierto. Esta librería, además de permitir manejar toda la funcionalidad SNMP a través de java, tiene la ventaja de tener preconfiguradas las MIB de tipo estándar y adicionalmente, permite la creación de MIB de tipo privado.

- Monitoreo de sensores: este módulo se implementa a través de un hilo que lanza un temporizador por cada sensor configurado, los cuales se ejecutan de acuerdo con el tiempo configurado en el archivo de configuración (ver parámetro tiempo en la etiqueta sensor en el ANEXO 1) y verifican el estado de los sensores a través de los atributos de los drivers de los sensores indicados en la Tabla 8.

Adicionalmente, a través de un temporizador que se ejecuta de manera periódica, se envía el mensaje de heartbeat al Gateway IoT, de acuerdo con el tiempo configurado en el archivo de configuración (ver parámetro heartbeat en el ANEXO 1).

Finalmente, en la Tabla 9 se presenta un resumen de los elementos del dispositivo IoT que fueron seleccionados.

Tabla 9. Resumen de elementos del dispositivo IoT

ELEMENTO	DESCRIPCIÓN	
SBC	Raspberry Pi Zero W • Procesador de un núcleo a 1GHz • 512 MB de RAM • Wifi 802.11n	
Sensores	hc-sr04 Sensor de ultrasonido	DS18B20 Sensor de temperatura y humedad
Sistema operativo	Raspbian	
Lenguaje de programación	• Java, para el desarrollo del agente, con sus diferentes módulos. • C, para el desarrollo de módulos de Kernel de Linux, para la comunicación con los sensores.	

Fuente: Elaboración propia

4.3 GATEWAY IOT

Para la implementación del dispositivo IoT, es importante tener en cuenta que existen en el mercado multiplicidad de SBC. La única condición requerida es que debe tener 2 interfaces de red: una para la comunicación LAN con los dispositivos IoT (que debe ser WiFi por lo indicado en la implementación del dispositivo IoT) y otra para la comunicación WAN con la aplicación de gestión.

Por razones de costo y disponibilidad inmediata[23] fue seleccionado una Intel NUC X3700m, la cual tiene un procesador de 2 núcleos a 1,8GHz, 4GB de memoria RAM y conectividad a través de Wifi 802.11n.

[22] https://www.snmp4j.org/
[23] La Intel NUC X3700m tuvo un costo de $300.000 a través de eBay en el año 2012.

Como sistema operativo fue seleccionado Ubuntu, que es una opción de uso libre.

Al igual que en el dispositivo IoT, los módulos del Gateway IoT se implementaron mediante el uso de hilos:

- Administración: A través de un hilo se crea el servidor Telnet, el cual a su vez lanza nuevos hilos por cada conexión establecida

- Gestor SNMP: para este módulo también se utilizó la librería SNMP4J, la cual como se indicó permite manejar toda la funcionalidad SNMP a través de java, para el caso del dispositivo IoT es especialmente importante ya que soporta la gestión de los traps enviados por los dispositivos IoT, permitiendo adicionalmente definir traps de tipo privado.

- Monitoreo de dispositivos IoT: este módulo se implementa a través de un hilo que lanza un temporizador por cada dispositivo IoT configurado del cual se recibo un mensaje de heartbeat, y de acuerdo con el tiempo consultado al dispositivo IoT a través del módulo Gestor SNMP.

- Agente SOAP: este módulo se implementa a través de 2 hilos, un cliente, mediante el cual se envían mensajes a la aplicación de gestión, y un servidor, mediante el cual se reciben y procesan los mensajes enviados por la aplicación de gestión. Los mensajes SOAP son implementados mediante el uso de la librería javax.jws que está incluida dentro de la distribución de java.

Adicionalmente, a través de un temporizador que se ejecuta de manera periódica, se envía el mensaje de heartbeat a la aplicación de gestión, de acuerdo con el tiempo configurado en el archivo de configuración (ver parámetro heartbeat en el ANEXO 2).

En la Tabla 10 se presenta un resumen de los elementos del Gateway IoT que fueron seleccionados.

Tabla 10. Resumen de elementos del Gateway IoT

ELEMENTO	DESCRIPCIÓN
SBC	Intel NUC X3700m • Procesador de 2 núcleos a 1,8GHz • 4GB de RAM • Wifi 802.11n (LAN) • Ethernet 100Mbps (WAN) • SSD de 32 GB
Sistema operativo	Ubuntu
Lenguaje de programación	Java, para el desarrollo del Gateway IoT, con sus diferentes módulos

Fuente: Elaboración propia

4.4 APLICACIÓN DE GESTIÓN

Para la implementación de la aplicación de gestión, es importante tener en cuenta que existen en el mercado multiplicidad de plataformas IaaS, y cualquiera puede ser susceptible de ser utilizada para la implementación del prototipo. Sin embargo, principalmente por razones de costo fue seleccionada la plataforma de DigitalOcean.

DigitalOcean es un proveedor estadounidense de IaaS, que ofrece servidores virtuales privados desde 5 dólares al mes, aunque el cobro se realiza por hora de uso del servidor, lo que lo hace ideal

en fases de desarrollo. Además, cuenta con servicios adicionales de Firewall, Backup, gestión de dominios, monitoreo de recursos, entre otros.

Como sistema operativo fue seleccionado Ubuntu, que es una de las opciones disponibles en DigitalOcean.

Para la implementación de la aplicación de gestión se utilizó un servidor de aplicaciones. Lo anterior, teniendo en cuenta que además de implementar la aplicación de gestión, esta tuviera una interfaz de acceso Web. Teniendo en cuenta que el lenguaje de programación es Java, se seleccionó el servidor de aplicaciones Apache Tomcat, el cual es de acceso gratuito y ampliamente utilizado. Como base de datos, se seleccionó MySQL, la cual es igualmente de amplia utilización y adicionalmente es gratuita.

A partir de lo anterior, los módulos de la aplicación de gestión se implementaron en el servidor de aplicaciones de la siguiente manera:

- Administración: se desarrolló una interfaz web mediante la cual se puede consultar y administración la información de los Gateway IoT, los dispositivos IoT y sus sensores.

- Agente SOAP: este módulo se implementa a través de 2 hilos, un servicio web, que actúa como servidor y mediante el cual se reciben y procesan los mensajes SOAP enviados por el Gateway IoT, y un cliente, mediante el cual se envían mensajes SOAP al Gateway IoT. Al igual que en el caso del Gateway IoT, los mensajes SOAP son implementados mediante el uso de la librería javax.jws que está incluida dentro de la distribución de java.

- Monitoreo de Gateway IoT: este módulo se implementa a través de un hilo que lanza un temporizador por cada Gateway IoT configurado, del cual se recibe un mensaje SOAP de heartbeat, y de acuerdo con el tiempo consultado al Gateway IoT a través del módulo Agente SOAP.

En la Tabla 11 se presenta un resumen de los elementos del Gateway IoT que fueron seleccionados.

Tabla 11. Resumen de elementos del Gateway IoT

ELEMENTO	**DESCRIPCIÓN**
IaaS	Cloud de DigitalOcean Droplet (servidor virtual privado). Para efectos de las pruebas realizadas, se seleccionó la opción más básica. Sin embargo, todos los recursos son escalables, según la necesidad: • 1 vCPU @ 1,7GHz • 1GB de RAM Además, cuenta con servicios adicionales de Firewall, Backup, gestión de dominios, monitoreo de recursos, entre otros
Sistema operativo	Ubuntu
Lenguaje de programación	Java, para el desarrollo de la aplicación de gestión, con sus 4 módulos (agente SOAP, Distribución de software, monitoreo de Gateway IoT y Administración)
Servidor de aplicaciones	Apache Tomcat
Base de datos	MySQL

Fuente: Elaboración propia

5 PRUEBAS Y ANÁLISIS DE RESULTADOS

En esta sección se presentarán las pruebas realizadas al prototipo implementado, con el fin de verificar y validar la función de gestión de fallos.

5.1 PROTOCOLO DE PRUEBAS

El presente protocolo de pruebas tiene como objetivo fundamental evaluar el funcionamiento de la solución diseñada, para lo cual se debe tener en consideración que este en un conjunto de componentes que interactúan entre sí y no un tipo de software individual que puede ser evaluado de forma independiente.

Así mismo, en el presente protocolo únicamente se desarrollan pruebas de los elementos de software desarrollados, obviando aquellas relacionadas con el hardware, en el entendido de que se asume la idoneidad de los elementos utilizados.

Con base en lo anterior, a continuación, se detallan las pruebas, la cuales están relacionadas con los procesos descritos anteriormente en la sección 3.6, y que están relacionados con la función de gestión de fallos.

5.1.1 Pruebas de inicialización de dispositivos

El objetivo es evaluar el proceso de inicialización de cada uno de los componentes del sistema de gestión de dispositivos IoT, es decir, dispositivo IoT, Gateway IoT y aplicación de gestión, con el fin de revisar su correcto funcionamiento.

5.1.1.1 Prueba de inicialización del dispositivo IoT

- **Prerrequisitos:** el archivo de configuración del dispositivo IoT debe tener la estructura correcta, la cual se muestra en el ANEXO 1.

- **Pasos:** para la realización de esta prueba, se deben ejecutar los siguientes pasos de manera satisfactoria:
  1. Inicio de los módulos del dispositivo IoT:
     a. Agente SNMP: En estado de espera de mensajes SNMP por parte del Gateway IoT, en el puerto configurado, y de acciones por parte de los demás módulos.
     b. Monitoreo de sensores: Lectura de los estados administrativo y operativo de cada uno de los sensores incluidos en la base de datos de configuración, de acuerdo con el valor de tiempo configurado.
     c. Administración: es espera de conexiones Telnet, en el puerto configurado.
     d. Adquisición de software: en espera de orden para descarga de archivo de software, desde el servidor SFTP configurado.
  2. Envío de un Trap de tipo *coldstart* hacia el Gateway IoT.
  3. Envío de Traps de tipo *heartbeat* hacia el Gateway IoT, de acuerdo con el valor de tiempo configurado.

- **Resultado esperado:** los resultados que se deben obtener son los siguientes:
  1. Todos los módulos del dispositivo IoT deben estar en ejecución, con excepción del módulo de "Adquisición de software", el cual, cómo se indicó anteriormente, no fue implementado en el prototipo.
  2. Envío de un trap de tipo coldstart al Gateway IoT previamente configurado.
  3. Envío de Traps sucesivos de tipo *heartbeat* hacia el Gateway IoT previamente configurado, de acuerdo con el valor de tiempo establecido.

5.1.1.2 Prueba de inicialización del Gateway IoT

- **Prerrequisitos:** el archivo de configuración del Gateway IoT debe tener la estructura correcta, la cual se muestra en el ANEXO 2.

- **Pasos:** para la realización de esta prueba, se deben ejecutar los siguientes pasos de manera satisfactoria:
  1. Inicio de los módulos del Gateway IoT:
     a. Gestor SNMP: En estado de espera de mensajes SNMP por parte del dispositivo IoT, en el puerto configurado, y de acciones por parte de los demás módulos.
     b. Agente SOAP: En estado de espera de mensajes SOAP por parte de la aplicación de gestión, y de acciones por parte de los demás módulos.
     c. Monitoreo de dispositivos IoT: En estado de espera de detección de dispositivos IoT configurados. Inicialmente, todos los dispositivos IoT configurados se clasifican como "no disponibles".
     d. Administración: es espera de conexiones Telnet, en el puerto configurado.
  2. Envío de un mensaje SOAP hacia la aplicación de gestión con la información de los dispositivos IoT configurados.
  3. Envío de mensajes SOAP de tipo *heartbeat* hacia la aplicación de gestión, de acuerdo con el valor de tiempo configurado.

- **Resultado esperado:** los resultados que se deben obtener son los siguientes:
  1. Todos los módulos del Gateway IoT deben estar en ejecución, con excepción del módulo de "Distribución de software", el cual, cómo se indicó anteriormente, no fue implementado en el prototipo.
  2. Envío de un mensaje SOAP a la aplicación de gestión previamente configurada, con la información de los dispositivos IoT asociados al Gateway IoT.
  3. Envío de mensajes SOAP sucesivos hacia la aplicación de gestión previamente configurada, de acuerdo con el valor de tiempo establecido.

5.1.1.3 Prueba de inicialización de la aplicación de gestión

- **Prerrequisitos:** la base de datos de la aplicación de gestión debe tener la estructura correcta, la cual se muestra en el ANEXO 3.

- **Pasos:** para la realización de esta prueba, se deben ejecutar los siguientes pasos de manera satisfactoria:
  1. Inicio de los módulos del gateway IoT:
     a. Agente SOAP: En estado de espera de mensajes SOAP por parte del Gateway IoT, y de acciones por parte de los demás módulos.
     b. Monitoreo de gateway IoT: En estado de espera de detección de gateway IoT configurados.
     c. Administración: es espera de conexiones Web, en el puerto configurado.

- **Resultado esperado:** los resultados que se deben obtener son los siguientes:
  1. Todos los módulos de la aplicación de gestión deben estar en ejecución, con excepción del módulo de "Distribución de software", el cual, cómo se indicó anteriormente, no fue implementado en el prototipo.

5.1.2 Prueba de detección de un dispositivo IoT parte del Gateway IoT

Esta prueba tiene como objetivo realizar la detección de un dispositivo IoT por parte del Gateway IoT, y su notificación a la aplicación de gestión.

5.1.2.1 Prueba de detección de un dispositivo IoT “no iniciado” parte del Gateway IoT

En esta prueba, se realiza la detección del dispositivo IoT mediante el trap de tipo *coldstart*

- **Prerrequisitos:** los prerrequisitos para realización de esta prueba son los siguientes:
  1. El Gateway IoT debe haber sido iniciado.
  2. La aplicación de gestión debe haber sido iniciada.
  3. El dispositivo IoT no debe haber sido iniciado.

- **Pasos:** para la realización de esta prueba, se deben ejecutar los siguientes pasos de manera satisfactoria:
  1. Inicializar el dispositivo IoT.
  2. El Gateway IoT recibe un trap de tipo *coldstart* por parte del dispositivo IoT.
  3. El Gateway IoT solicita la configuración al dispositivo IoT sobre:
     a. Los OID del subárbol system.
     b. Los OID del subárbol de sensores IoT
     c. La versión de software. (Si la versión de software es inferior a la requerida, solicita al dispositivo IoT su actualización. Este proceso es explicado más adelante).
     d. El OID de heartbeat
  4. El Gateway IoT actualiza su base de datos de configuración, cambiando el estado del dispositivo IoT a “disponible”.
  5. El Gateway IoT inicializa el monitoreo del dispositivo IoT, a través del establecimiento de un temporizador basado en el valor del OID de heartbeat.
  6. El Gateway IoT actualiza la información del dispositivo IoT hacia la aplicación de gestión.
  7. La aplicación de gestión actualiza su base de datos de configuración.

- **Resultado esperado:** los resultados que se deben obtener son los siguientes:
  1. El dispositivo IoT debe ser identificado como disponible en el Gateway IoT.
  2. El dispositivo IoT debe ser identificado como disponible en la aplicación de gestión.

5.1.2.2 Prueba de detección de un dispositivo IoT “iniciado” parte del Gateway IoT

En esta prueba, se realiza la detección del dispositivo IoT mediante el trap de tipo *heartbeat*.

- **Prerrequisitos:** los prerrequisitos para realización de esta prueba son los siguientes:
  1. El Gateway IoT no debe haber sido iniciado.
  2. La aplicación de gestión debe haber sido iniciada.
  3. El dispositivo IoT debe haber sido iniciado.

- **Pasos:** para la realización de esta prueba, se deben ejecutar los siguientes pasos de manera satisfactoria:
  1. Inicializar el Gateway IoT.
  2. El Gateway IoT recibe un trap de tipo *heartbeat* por parte del dispositivo IoT.
  3. El Gateway IoT solicita la configuración al dispositivo IoT sobre:
     a. Los OID del subárbol system.
     b. Los OID del subárbol de sensores IoT
     c. La versión de software. (Si la versión de software es inferior a la requerida, solicita al dispositivo IoT su actualización. Este proceso es explicado más adelante).
     d. El OID de heartbeat
  4. El Gateway IoT actualiza su base de datos de configuración, cambiando el estado del dispositivo IoT a “disponible”.

5. El Gateway IoT inicializa el monitoreo del dispositivo IoT, a través del establecimiento de un temporizador basado en el valor del OID de heartbeat.
6. El Gateway IoT actualiza la información del dispositivo IoT hacia la aplicación de gestión.
7. La aplicación de gestión actualiza su base de datos de configuración.

- **Resultado esperado:** los resultados que se deben obtener son los siguientes:
  1. El dispositivo IoT debe ser identificado como disponible en el Gateway IoT.
  2. El dispositivo IoT debe ser identificado como disponible en la aplicación de gestión.

5.1.3 Prueba de determinación de un dispositivo IoT como "no disponible"

El objetivo es evaluar el proceso de determinar un dispositivo IoT como no disponible por parte del Gateway IoT, y su notificación a la aplicación de gestión.

- **Prerrequisitos:** los prerrequisitos para realización de esta prueba son los siguientes:
  1. El Gateway IoT debe haber sido iniciado.
  2. La aplicación de gestión debe haber sido iniciada.
  3. El dispositivo IoT debe haber sido iniciado.
  4. El dispositivo IoT debe haber sido detectado como disponible.

- **Pasos:** para la realización de esta prueba, se deben ejecutar los siguientes pasos de manera satisfactoria:
  1. Apagar el dispositivo IoT.
  2. El contador de heatbeat del Gateway IoT asociado al dispositivo IoT debe llegar a "0".
  3. El Gateway IoT identifica al dispositivo IoT como no disponible.
  4. El Gateway IoT envía un mensaje SOAP a la aplicación de gestión para informar sobre el cambio de estado del dispositivo IoT.
  5. La aplicación de gestión recibe el mensaje SOAP y actualiza su base de datos de configuración.

- **Resultado esperado:** los resultados que se deben obtener son los siguientes:
  1. El dispositivo IoT debe ser identificado como no disponible en el Gateway IoT.
  2. El dispositivo IoT debe ser identificado como no disponible en la aplicación de gestión.

5.1.4 Prueba de detección de un Gateway IoT por parte de la aplicación de gestión:

Esta prueba tiene como objetivo realizar la detección de un Gateway IoT por parte de la aplicación de gestión.

- **Prerrequisitos:** los prerrequisitos para realización de esta prueba son los siguientes:
  1. El Gateway IoT no debe haber sido iniciado.
  2. La aplicación de gestión debe haber sido iniciada.
  3. El dispositivo IoT debe haber sido iniciado.

- **Pasos:** para la realización de esta prueba, se deben ejecutar los siguientes pasos de manera satisfactoria:

  1. Inicializar el Gateway IoT.
  2. Gateway IoT envía un mesaje SOAP de heartbeat a la aplicación de gestión.
  3. La aplicación de gestión identifica que el gateway IoT no ha sido detectado, consultando su base de datos de configuración.
  4. La aplicación de gestión actualiza su base de datos de configuración, cambiando el estado del Gateway IoT a "disponible".

5. La aplicación de gestión solicita la información de los dispositivos IoT conectados al Gateway IoT:
   a. Los OID del subárbol system.
   b. Los OID del subárbol de sensores IoT
   c. La versión de software
6. El Gateway IoT envía un mensaje SOAP con la información solicitada.
7. La aplicación de gestión recibe el mensaje SOAP y actualiza su base de datos
8. La aplicación de gestión inicializa el monitoreo del Gateway IoT, a través del establecimiento de un temporizador.

- **Resultado esperado:** los resultados que se deben obtener son los siguientes:
  1. El Gateway IoT debe ser identificado como disponible en la aplicación de gestión.

5.1.5 Prueba de determinación de un Gateway IoT como "no disponible"

El objetivo de esta prueba es evaluar el proceso de determinar un Gateway IoT como no disponible por parte de la aplicación de gestión.

- **Prerrequisitos:** los prerrequisitos para realización de esta prueba son los siguientes:
  1. La aplicación de gestión debe haber sido iniciada.
  2. El Gateway IoT debe haber sido iniciado.
  3. El Gateway IoT debe haber sido detectado como disponible.

- **Pasos:** para la realización de esta prueba, se deben ejecutar los siguientes pasos de manera satisfactoria:
  1. Apagar el Gateway IoT.
  2. El contador de heatbeat de la aplicación de gestión asociado al Gateway IoT debe llegar a "0".
  3. La aplicación de gestión identifica al Gateway IoT como no disponible.
  4. La aplicación de gestión actualiza su base de datos de configuración.

- **Resultado esperado:** los resultados que se deben obtener son los siguientes:
  1. El Gateway IoT debe ser identificado como no disponible en la aplicación de gestión.

5.1.6 Prueba de detección de cambio en el estado de un sensor del dispositivo IoT

Su objetivo es el de evaluar el proceso de detección de cambios de estados de los sensores del dispositivo IoT por parte gel Gateway IoT, y su notificación a la aplicación de gestión.

- **Prerrequisitos:** los prerrequisitos para realización de esta prueba son los siguientes:
  1. La aplicación de gestión debe haber sido iniciada.
  2. El Gateway IoT debe haber sido iniciado y detectado como disponible.
  3. El dispositivo IoT debe haber sido iniciado y detectado como disponible.
  4. El estado operativo del sensor del dispositivo IoT debe ser ACTIVADO.

- **Pasos:** para la realización de esta prueba, se deben ejecutar los siguientes pasos de manera satisfactoria:
  1. Desconectar el sensor del dispositivo IoT
  2. El dispositivo detecta el cambio de estado operativo del sensor, a DESACTIVADO.
  3. El dispositivo IoT envía un trap de tipo *sensorDownOper* al Gateway IoT.
  4. El Gateway IoT recibe el trap y actualiza su base de datos de configuración
  5. El Gateway IoT envía un mensaje SOAP a la aplicación de gestión, informando sobre el cambio en el estado un sensor.

6. La aplicación de gestión recibe el mensaje SOAP y actualiza su base de datos de configuración.

- **Resultado esperado:** los resultados que se deben obtener son los siguientes:
  1. El estado operador del sensor del dispositivo IoT debe ser identificado como DESACTIVADO en el Gateway IoT.
  2. El estado operador del sensor del dispositivo IoT debe ser identificado como DESACTIVADO en la aplicación de gestión.

5.2 ANÁLISIS Y RESULTADOS

En el ANEXO 4 se presentan los resultados obtenidos de las pruebas realizadas y que fueron descritas en la sección anterior. Sin embargo, a manera de ejemplo, en la presente sección se presenta el desarrollo únicamente de una de ellas.

A continuación, se presentan los resultados obtenidos de las pruebas realizadas y que fueron descritas en la sección anterior.

Para obtener los resultados se utilizaron las siguientes herramientas:
- Wireshark, que es un analizador de protocolos;
- PuTTY, que es un cliente Telnet y SSH.
- Salidas impresas en consola por el dispositivo IoT y el Gateway IoT, y la interfaz web de la aplacación de gestión.

5.2.1 Resultado de prueba de inicialización del dispositivo IoT

Cuando se inicializa el dispositivo IoT, este imprime en la consola la siguiente información:

```
------------------------------------------------
[CONFIGURACION]
address: 0.0.0.0
versionSNMP: 2c
SnmpPort: 161
ConsolePort: 10578
Heartbeat: 2 seg
SysName: Agente IoT
SysLocation: Fuera
SysContact: Miguel Duran
SFTP: maduran:123456@192.168.2.107:22 folder:/home/maduran/IoT/
Sensor: ULTRASONIDO
  name: HC-SR04
  driver: hc_sr04
  ruta: /sys/class/hc_sr04/distance_17_27/
  tiempo monitoreo: 1 seg
Sensor: TEMPERATURA
  name: DS18B20
  driver: w1_therm
  ruta: /sys/bus/w1/devices/28-0417616eecff/
  tiempo monitoreo: 2 seg
Community: public/rw
Manager (manager 1): udp:192.168.2.102/10000
------------------------------------------------
[INICIO]
Agente SNMP [OK]
Sensores [OK]
Cliente SFTP [OK]
Consola [OK]
```

En primer lugar, el dispositivo IoT presenta la información del archivo de configuración, en la que se puede observar la configuración SNMP relacionada con el puerto SNMP, el Gateway IoT (manager), la comunidad SNMP, así como la información de los sensores configurados en el dispositivo.

Posteriormente, el dispositivo IoT presenta la información de inicialización de los diferentes módulos que hacen parte de él:

- La inicialización del módulo de Administración se verifica mediante PuTTY, estableciendo una conexión Telnet a la IP del dispositivo IoT y al puerto 10578, que es el puerto configurado. El resultado se a continuación.

Adicionalmente, en la consola del dispositivo IoT se muestra un mensaje cuando se detecta una nueva conexión Telnet.

```
Consola: Nueva conexión entrante: Socket[addr=/192.168.2.103,port=52361,localport=10578]
```

- La inicialización del módulo de Agente SNMP se verifica con Wireshark, en el que se pueden observar los mensajes SNMP que envía el dispositivo IoT al Gateway IoT. En este, además se puede evidenciar el envío del trap *coldstart* (OID 1.3.6.1.6.3.1.1.5.1) y de los traps de tipo *heartbeat* sucesivos (OID 1.3.6.1.6.3.1.1.5.50000.1).

No.	Time	Source	Destination	Protocol	Length	Info
2249	17.447060	192.168.2.114	192.168.2.102	SNMP	87	snmpV2-trap 1.3.6.1.6.3.1.1.5.1
2251	17.461681	192.168.2.114	192.168.2.102	SNMP	90	snmpV2-trap 1.3.6.1.6.3.1.1.5.50000.1
2272	19.468970	192.168.2.114	192.168.2.102	SNMP	90	snmpV2-trap 1.3.6.1.6.3.1.1.5.50000.1
2279	21.458289	192.168.2.114	192.168.2.102	SNMP	90	snmpV2-trap 1.3.6.1.6.3.1.1.5.50000.1
2291	23.458964	192.168.2.114	192.168.2.102	SNMP	90	snmpV2-trap 1.3.6.1.6.3.1.1.5.50000.1
2304	25.459890	192.168.2.114	192.168.2.102	SNMP	90	snmpV2-trap 1.3.6.1.6.3.1.1.5.50000.1
2313	27.459706	192.168.2.114	192.168.2.102	SNMP	90	snmpV2-trap 1.3.6.1.6.3.1.1.5.50000.1

- La inicialización del módulo de Monitoreo de sensores, además de que el dispositivo IoT lo muestra en consola, se puede verificar adicionalmente realizando la desconexión de un sensor, con lo cual se muestra el siguiente mensaje.

```
[Sensores] 2019/05/05 12:53:35
HC-SR04: inactivo (Operativo)
```

6 CONCLUSIONES

El desarrollo del presente trabajo permitió identificar y analizar las principales iniciativas de sistemas de gestión de red para redes IoT, en las cuales se existen cuatro requisitos fundamentales de gestión de dispositivos para cualquier despliegue de dispositivos de IoT relacionados con el aprovisionamiento y autenticación, configuración y control, monitoreo y diagnóstico, y

actualizaciones de software y mantenimiento. Adicionalmente, se identificó yqe existen 4 tendencias relacionadas con las formas de implementación: SNMP, TMN, WBEM y PBM.

Por otra parte, se realizó la especificación y diseño de la arquitectura del sistema de gestión de red de una red IoT, incluyendo sus elementos: dispositivo IoT, Gateway IoT y nube, y bajo el modelo de computación en el borde. Este diseño permite administrar de manera simple los dispositivos desplegados en una red IoT. La estructura del modelo de objetos de información diseñado permite gestionar de manera sencilla la información de los sensores de los dispositivos IoT.

Es importante tener en cuenta que en escenarios donde existan miles de dispositivos IoT, las tasas de transferencia pueden ver afectadas si no se configuran de manera adecuada los temporizadores de monitoreo de los mismos, es decir, se establecen en valores muy bajos, teniendo en cuenta que estos de manera periódica envían mensajes de heartbeat al Gateway IoT. Así mismo, en redes locales inestables, los dispositivos IoT pueden ser detectados de manera constante como no disponibles por el Gateway IoT, lo que afectaría las tasas de transferencia entre este elemento y la aplicación de gestión, pues constantemente se estarían enviando mensaje SOAP informando sobre estos cambios de estado.

De otro lado, a pesar de que en el diseño desarrollado únicamente se contemplaron características de los sensores asociadas a sus estados (administrativo y operativo), la estructura de objetos de información propuesta permite a futuro agregar características adicionales de los sensores o del dispositivo IoT en sí mismo, como por ejemplo las relacionadas con consumo energético.

Por otra parte, la heterogeneidad de las características de los dispositivos y sensores existentes en el mercado, pueden ser homogeneizadas a través del sistema de gestión de red diseñado, es decir, por ejemplo, que mediante la estandarización de la forma de consultar el estado (administrativo y operativo) de los sensores, se facilitan las labores de gestión de los mismo. No obstante, esto implica el desarrollo de un driver para cada sensor de manera específica.

Uno de los elementos esenciales del modelo de computación en el borde es el Gateway IoT. En el diseño desarrollado se da especial importante a este elemento, pues es el que realiza el monitoreo constante de los dispositivos IoT, aumentado la confiabilidad del sistema de gestión pues se elimina la dependencia absoluta de enlaces hacia la nube.

El diseño desarrollado, al hacer uso de protocolos como SNMP y SOAP permite apoyar el proceso de estandarización, pues estos son ampliamente utilizados actualmente en la industria.

En cuanto al prototipo implementado, este permitió realizar la validación del funcionamiento de las características diseñadas, a través de un protocolo de pruebas desarrollado para tal fin. Teniendo en cuenta que la implementación fue realizada en su mayoría mediante lenguaje java, a futuro se puede evaluar la misma mediante el uso del lenguaje C, específicamente para el dispositivo IoT y el Gateway IoT.

Los usuarios potenciales del prototipo implementado son aquellas empresas que tengan implementada una red de dispositivos IoT y necesiten una herramienta que permita realizar la administración de estos.

Finalmente, los futuros desarrollos en el sistema de gestión de red diseñado pueden estar relacionados con la implementación de SNMP v3 a nivel de la comunicación entre el dispositivo IoT y el Gateway IoT. Así mismo, en cuanto a la comunicación entre el Gateway IoT y la aplicación de gestión, a pesar de que el protocolo seleccionado fue SOAP, se podrían implementar protocolos adicionales como REST, permitiéndoles al usuario su selección, dependiente de las características de desempeño que necesite de manera particular.

7 BIBLIOGRAFÍA

[1] Jadoul, M. (2015). The IoT: The next step in internet evolution. Nokia. [En línea]. [citado 12 de noviembre de 2016], Formato HTML, Disponible en Internet: https://insight.nokia.com/iot-next-step-internet-evolution

[2] Recomendación UIT-T Y.2060. Descripción general de Internet de los objetos (2012). Unión Internacional de las Telecomunicaciones.

[3] Ericsson (2018). Ericsson Mobility Report November 2018. [En línea]. [citado 19 de mayo de 2019], Formato PDF, Disponible en Internet: https://www.ericsson.com/assets/local/mobility-report/documents/2018/ericsson-mobility-report-november-2018.pdf

[4] Statista. (2015). Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions). Statista. [En línea]. [citado 19 de mayo de 2019], Formato HTML, Disponible en Internet: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[5] Evans, D. (2011). Internet de las cosas. Cómo la próxima evolución de Internet lo cambia todo. Cisco Internet Business Solutions Group.

[6] Cisco Global Cloud Index. Forecast and Methodology, 2015-2020 (2014).

[7] Vaquero, D. & Rodero-Merino, L. (2014). Finding your Way in the Fog: Towards a Comprehensive Definition of Fog Computing.

[8] Yi, S., Hao, Z., Qin, Z. & Li, Q. (2015). Fog Computing: Platform and Applications. Third IEEE Workshop on Hot Topics in Web Systems and Technologies. Computer Science and Information Systems.

[9] Bonomi, F., Milito, R., Natarajan, P. & Zhu, J. (2014). Fog Computing: A Platform for Internet of Things and Analytics. Springer International Publishing.

[10] Bonomi, F., Milito, R., Zhu, J. & Addepalli, S. (2012). Fog Computing and Its Role in the Internet of Things. Cisco Systems Inc.

[11] Dastjerdi, A., Gupta, J., Calheiros, R., Ghosh, S & Buyya, R. (2016). Internet of Things. Principles and Paradigms. Chapter 4. Fog Computing: Principles, Architectures, and Applications. ELSEVIER.

[12] Hu, P., Ning, H. Qiu, T., Zhang, Y. & Luo, X. (2016). Fog Computing-Based Face Identification and Resolution Scheme in Internet of Things. IEEE.

[13] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. IEEE Internet of Things Journal, Vol. 3, No. 5.

[14] Martí, A. (1999) Gestión de redes. Universidad de Cataluña.

[15] Recomendación UIT-T X.701. (1997) Gestión de interconexión de sistemas abiertos - Marco y arquitectura de la gestión de sistemas. Unión Internacional de Telecomunicaciones.

[16] Romero, M. Gestión de Redes. Sistemas Avanzados de Comunicaciones. Universidad de Sevilla. [En línea]. [citado 10 de febrero de 2017], Formato PDF, Disponible en Internet:

http://www.dte.us.es/personal/mcromero/docs/sac/sac-gestionderedes.pdf

[17] Recomendación UIT-T M.3010. (2000) Principios para una red de gestión de las telecomunicaciones. Unión Internacional de Telecomunicaciones.

[18] Weber, J. Fundamentals of IoT device management. Avnet. [En línea]. [citado el 8 de abril de 2017], Formato HTML, Disponible en Internet: http://iotdesign.embedded-computing.com/articles/fundamentals-of-iot-device-management/

[19] Rodríguez, D. (2013). Arquitectura y Gestión de la IoT. Revista Telem@tica Vol. 12, No. 3 (pp. 49-60). Instituto Superior Politécnico José Antonio Echeverría.

[20] Hui-Ping, H., Shi-De, X., & Xiang-Yin, M. (2015). Applying SNMP Technology to Manage the Sensors in Internet of Things. The Open Cybernetics & Systemics Journal (pp. 1019-1024).

[21] Choi, H., Kim, N., & Cha, H. (2009). 6LoWPAN-SNMP: Simple network management protocol for 6LoWPAN. 11th IEEE International Conference on High Performance Computing and Communications (pp. 305-313). IEEE.

[22] Subramanian, M. (2000). Network Management: Principles and Practices. Addison Wesley.

[23] Luque, J., Gonzalo, F., Escudero, J. & Carrasco, A. (1999). TMN versus SNMP-based network management systems. Universidad de Sevilla. [En línea]. [citado el 23 de abril de 2017], Formato DF, Disponible en Internet: http://personal.us.es/jluque/Congresos/1999%20Cigre%20Polonia-1.pdf

[24] Foreman, J. (1997). Software Technology Review. Carnegie Mellon University. [En línea]. [citado el 1 de marzo de 2017], Formato PDF, Disponible en Internet: http://teaching.shu.ac.uk/aces/pc/LEC_NOTE/CSSD/DistributedObjectsRefernces/sei_softwaretechnologyreview.pdf

[25] A guide to understanding SNMP. (2013). SolarWinds Worldwide, LLC.

[26] Mauro, D. & Schmidt, K. (2005) Essential SNMP. Segunda edición. O'Reilly.

[27] Perkins, D. (1993) Understanding SNMP MIBs. Revision 1.1.7.

[28] Router Teldat. Agente SNMP. (2000) Doc. DM512 Rev. 8.40. [En línea]. [citado el 22 de abril de 2017], Formato PDF, Disponible en Internet: http://www.it.uc3m.es/~teldat/Cbra/castellano/protocolos/Dm512v840_Agente_SNMP.PDF

ANEXO 1. ESTRUCTURA DEL ARCHIVO DE CONFIGURACIÓN DEL DISPOSITIVO IOT

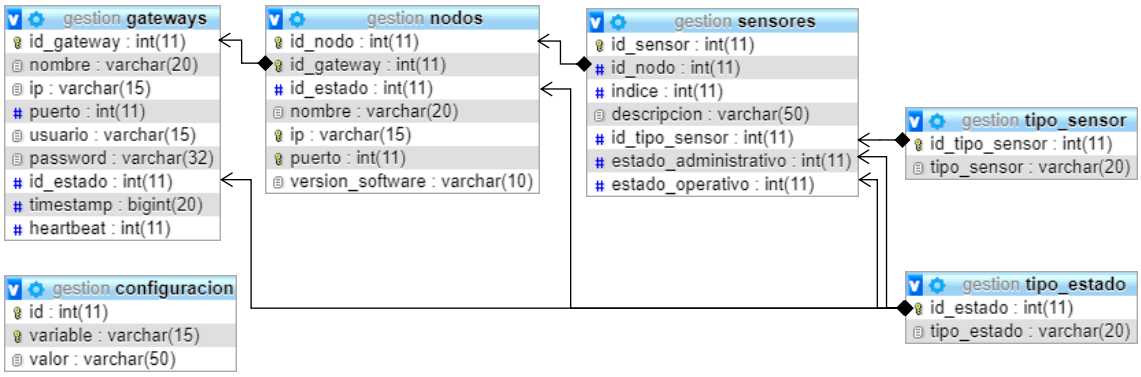
```
<?xml version="1.0" encoding="UTF-8"?>
<config>
  <auth>
    <user>root</user>
    <password>827ccb0eea8a706c4c34a16891f84e7b</password>
  </auth>
  <system>
    <sysName>Agente IoT</sysName>
    <sysLocation>Fuera</sysLocation>
    <sysContact>Miguel Duran</sysContact>
  </system>
  <remoteSFTP>
    <host>192.168.2.107</host>
    <port>22</port>
    <username>maduran</username>
    <password>575002</password>
    <folder>/home/maduran/IoT/</folder>
  </remoteSFTP>
  <address>0.0.0.0</address>
  <versionSNMP>2c</versionSNMP>
  <snmpPort>161</snmpPort>
  <consolePort>10578</consolePort>
  <heartbeat>2</heartbeat>
  <sensores>
    <sensor>
      <type>3</type>
      <name>HC-SR04</name>
      <driver>hc_sr04</driver>
      <ruta>/sys/class/hc_sr04/distance_17_27/</ruta>
      <tiempo>1</tiempo>
    </sensor>
    <sensor>
      <type>1</type>
      <name>DS18B20</name>
      <driver>w1_therm</driver>
      <ruta>/sys/bus/w1/devices/28-0417616eecff/</ruta>
      <tiempo>2</tiempo>
    </sensor>
  </sensores>
  <communities>
    <community>
      <name>public</name>
      <type>rw</type>
    </community>
  </communities>
  <managers>
    <manager>
      <name>manager 1</name>
      <protocol>udp</protocol>
      <address>192.168.2.102</address>
      <port>10000</port>
    </manager>
  </managers>
</config>
```

ANEXO 2. ESTRUCTURA DEL ARCHIVO DE CONFIGURACIÓN DEL GATEWAY IOT

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
  <auth>
    <user>root</user>
    <password>827ccb0eea8a706c4c34a16891f84e7b</password>
  </auth>
  <address>0.0.0.0</address>
  <ipPublica>192.168.2.105</ipPublica>
  <soap_port>8888</soap_port>
  <versionSNMP>2c</versionSNMP>
  <trapPort>10000</trapPort>
  <consolePort>10570</consolePort>
  <heartbeat>2</heartbeat>
  <aplicacion>
    <address>192.168.2.105</address>
    <port>7777</port>
    <user>root</user>
    <password>827ccb0eea8a706c4c34a16891f84e7b</password>
  </aplicacion>
  <agents>
    <agent>
      <name>agent 1</name>
      <protocol>udp</protocol>
      <address>192.168.2.105</address>
      <port>161</port>
      <community>public</community>
    </agent>
    <agent>
      <name>agent 2</name>
      <protocol>udp</protocol>
      <address>192.168.2.114</address>
      <port>161</port>
      <community>public</community>
    </agent>
    <agent>
      <name>agent 3</name>
      <protocol>udp</protocol>
      <address>192.168.2.117</address>
      <port>161</port>
      <community>public</community>
    </agent>
  </agents>
</config>
```

ANEXO 3. ESTRUCTURA DE LA BASE DE DATOS DE LA APLICACIÓN DE GESTIÓN

A continuación, se muestra la estructura de tablas y relaciones de la base de datos de la aplicación de gestión.



Las características de las tablas son las siguientes:

- **gateways:** información de los Gateway IoT
- **nodos:** información de los dispositivos IoT
- **sensores:** información de los sensores de los dispositivos IoT
- **tipo_sensores:** información sobre los tipos de sensores (temperatura / ultrasonido / etc.)
- **tipo_estado:** información sobre los tipos de estaods (desconectado / conectado)
- **configuración:** información de configuración de la aplicación de gestión, como el puerto SOAP, usuario, contraseña.

ANEXO 4. RESULTADOS DE LAS PRUEBAS

En el presente anexo se presentan los resultados de las pruebas realizadas al prototipo implementado.

RESULTADO DE LAS PRUEBAS DE INICIALIZACIÓN DE DISPOSITIVOS

Resultado de prueba de inicialización del dispositivo IoT

Cuando se inicializa el dispositivo IoT, este imprime en la consola la siguiente información:

```
-------------------------------------------------
[CONFIGURACION]
address: 0.0.0.0
versionSNMP: 2c
SnmpPort: 161
ConsolePort: 10578
Heartbeat: 2 seg
SysName: Agente IoT
SysLocation: Fuera
SysContact: Miguel Duran
SFTP: maduran:575002@192.168.2.107:22 folder:/home/maduran/IoT/
Sensor: ULTRASONIDO
  name: HC-SR04
  driver: hc_sr04
  ruta: /sys/class/hc_sr04/distance_17_27/
  tiempo monitoreo: 1 seg
Sensor: TEMPERATURA
  name: DS18B20
  driver: w1_therm
  ruta: /sys/bus/w1/devices/28-0417616eecff/
  tiempo monitoreo: 2 seg
Community: public/ro
Community: private/rw
Manager (manager 1): udp:192.168.2.102/10000
-------------------------------------------------
[INICIO]
Agente SNMP [OK]
Sensores [OK]
Cliente SFTP [OK]
Consola [OK]
```

En primer lugar, el dispositivo IoT presenta la información del archivo de configuración, en la que se puede observar la configuración SNMP relacionada con el puerto SNMP, el Gateway IoT (manager), la comunidad SNMP, así como la información de los sensores configurados en el dispositivo.

Posteriormente, el dispositivo IoT presenta la información de inicialización de los diferentes módulos que hacen parte de él:

- La inicialización del módulo de Administración se verifica mediante PuTTY, estableciendo una conexión Telnet a la IP del dispositivo IoT y al puerto 10578, que es el puerto configurado. El resultado se a continuación.

Adicionalmente, en la consola del dispositivo IoT se muestra un mensaje cuando se detecta una nueva conexión Telnet.

```
Consola: Nueva conexión entrante: Socket[addr=/192.168.2.103,port=52361,localport=10578]
```

- La inicialización del módulo de Agente SNMP se verifica con Wireshark, en el que se pueden observar los mensajes SNMP que envía el dispositivo IoT al Gateway IoT. En este, además se puede evidenciar el envío del trap *coldstart* (OID 1.3.6.1.6.3.1.1.5.1) y de los traps de tipo *heartbeat* sucesivos (OID 1.3.6.1.6.3.1.1.5.50000.1).

No.	Time	Source	Destination	Protocol	Length	Info
2249	17.447060	192.168.2.114	192.168.2.102	SNMP	87	snmpV2-trap 1.3.6.1.6.3.1.1.5.1
2251	17.461681	192.168.2.114	192.168.2.102	SNMP	90	snmpV2-trap 1.3.6.1.6.3.1.1.5.50000.1
2272	19.468970	192.168.2.114	192.168.2.102	SNMP	90	snmpV2-trap 1.3.6.1.6.3.1.1.5.50000.1
2279	21.458289	192.168.2.114	192.168.2.102	SNMP	90	snmpV2-trap 1.3.6.1.6.3.1.1.5.50000.1
2291	23.458964	192.168.2.114	192.168.2.102	SNMP	90	snmpV2-trap 1.3.6.1.6.3.1.1.5.50000.1
2304	25.459890	192.168.2.114	192.168.2.102	SNMP	90	snmpV2-trap 1.3.6.1.6.3.1.1.5.50000.1
2313	27.459706	192.168.2.114	192.168.2.102	SNMP	90	snmpV2-trap 1.3.6.1.6.3.1.1.5.50000.1

- La inicialización del módulo de Monitoreo de sensores, además de que el dispositivo IoT lo muestra en consola, se puede verificar adicionalmente realizando la desconexión de un sensor, con lo cual se muestra el siguiente mensaje.

```
 [Sensores] 2019/05/05 12:53:35
HC-SR04: inactivo (Operativo)
```

Resultado de prueba de inicialización del Gateway IoT

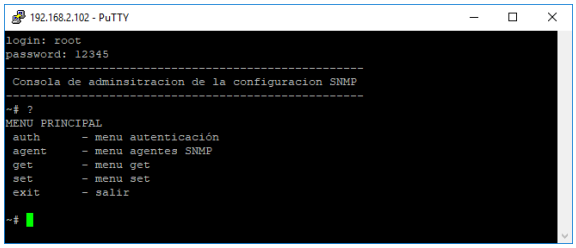
Cuando se inicializa el Gateway IoT, este imprime en la consola la siguiente información:

```
------------------------------------------------
[CONFIGURACIÓN]
address: 0.0.0.0
Dominio/IP pública: 192.168.2.105
SOAP port: 8888
versionSNMP: 2c
TrapPort: 10000
ConsolePort: 10570
Aplicacion: 192.168.2.105:7777
Agent (agent 1): udp/192.168.2.105:161
Agent (agent 2): udp/192.168.2.114:161
------------------------------------------------
[INICIO]
Gestor SNMP    [OK]
Consola        [OK]
Servidor SOAP  [OK]
Cliente SOAP   [OK]
```

En primer lugar, el Gateway IoT presenta la información del archivo de configuración, en la que se puede observar la configuración SNMP relacionada con el puerto SNMP Trap, los dispositivos IoT configurados (agents), la aplicación de gestión, entre otra información.

Posteriormente, el Gateway IoT presenta la información de inicialización de los diferentes módulos que hacen parte de él:

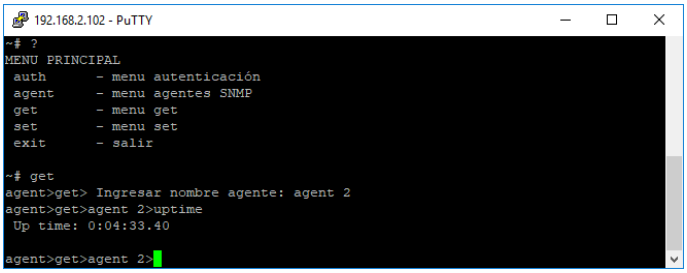
- La inicialización del módulo de Administración se verifica mediante PuTTY, estableciendo una conexión Telnet a la IP del dispositivo IoT y al puerto 10570, que es el puerto configurado. El resultado se a continuación.



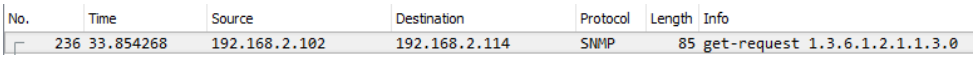
Adicionalmente, en la consola del dispositivo IoT se muestra un mensaje cuando se detecta una nueva conexión Telnet.

```
Consola: Nueva conexión entrante: Socket[addr=/192.168.2.103,port=53199,localport=10570]
```

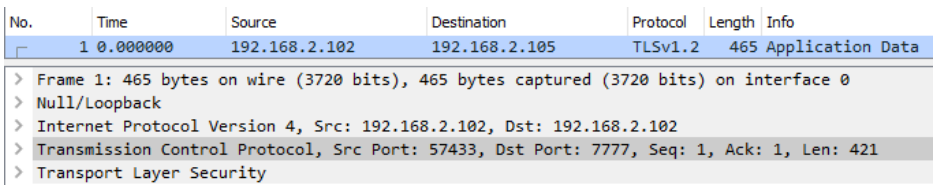
- La inicialización del módulo de Gestor SNMP, además de que el Gateway IoT lo muestra en consola, se puede verificar adicionalmente enviando un mensaje GetRequest a través del módulo de administración.



El mensaje enviado se puede observar en Wireshark



- La inicialización del módulo Agente SOAP, además de que el Gateway IoT lo muestra en consola, se puede verificar adicionalmente a través de Wireshark, en el que se observa el envío de los mensajes SOAP a la IP y puerto de la aplicación de gestión.



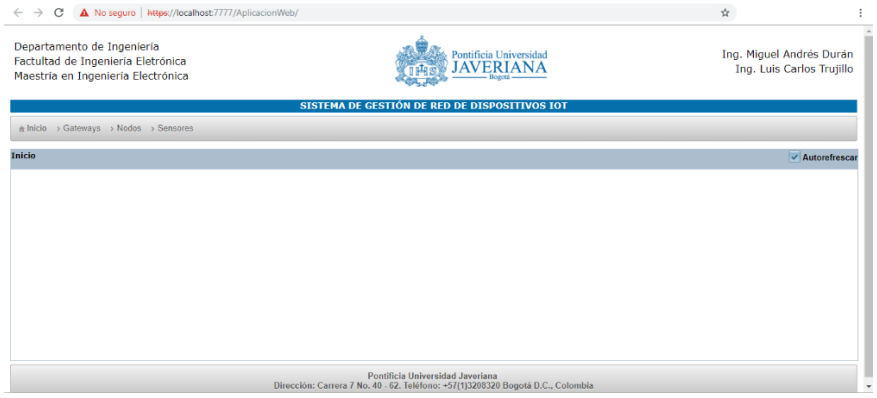
Resultado de prueba de inicialización de la aplicación de gestión

En primer lugar, la aplicación de gestión presenta la información de su base de datos de configuración, en la que se pueden observar los Gateway IoT configurados.

```
------------------------------------------------
[CONFIGURACIÓN]
SOAP Port: 7777
Gateway (gestor 1): 192.168.2.105:8888
Gateway (gestor 2): 192.168.2.102:7777

------------------------------------------------
[INICIO]
Aplicación     [OK]
Cliente SOAP   [OK]
```

Una vez inicializada la aplicación de gestión, es posible acceder mediante un navegador Web, a la url https://ip:puerto/, y se muestra el módulo de Administración:



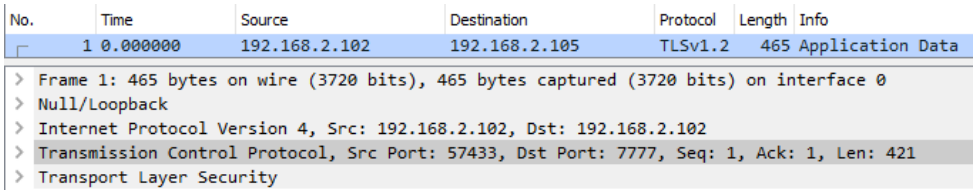
RESULTADOS DE LA PRUEBA DE DETECCIÓN DE UN DISPOSITIVO IOT PARTE DEL GATEWAY IOT

Detección de un dispositivo IoT “no iniciado” parte del Gateway IoT

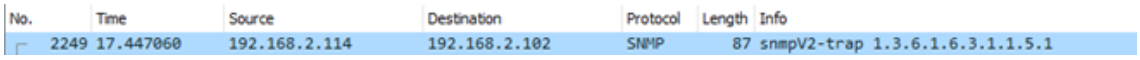
Luego de iniciado el dispositivo IoT, este envía un trap de tipo *coldstart* al Gateway IoT, el cual lo detecta y lo muestra en consola, y clasifica al dispositivo IoT como “disponible”, tal y como se muestra a continuación.

```
------------------------------------------------
[SNMP] 2019/05/19 13:15:49
TRAP
Agente: 192.168.2.114/161
OID: 1.3.6.1.6.3.1.1.5.1
Type: ColdStart
------------------------------------------------
[SNMP] 2019/05/05 13:15:49
Agente: 192.168.2.114/161
Estado: [disponible]
------------------------------------------------
[SOAP]
Agente: 192.168.2.114/161
Resultado: Nodo actualizado
------------------------------------------------
```

Posteriormente, el Gateway IoT envía un mensaje SOAP a la aplicación de gestión, informando de la detección del dispositivo IoT, lo cual se puede evidenciar a través de Wireshark.

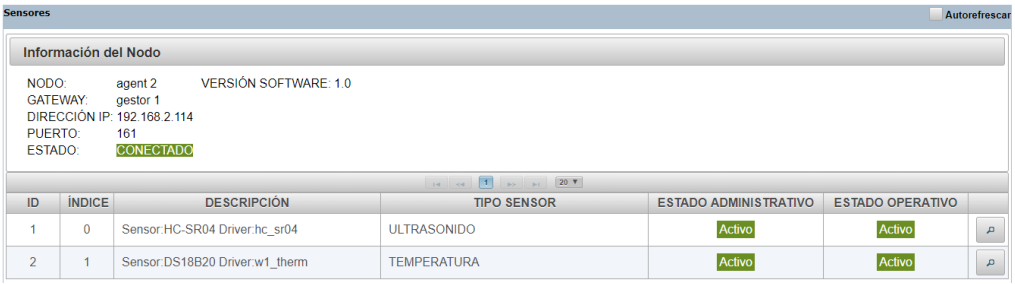


Finalmente, la aplicación de gestión recibe el mensaje SOAP y actualiza su base de datos de gestión, lo cual puede observarse a través del módulo de administración, en donde se evidencia que el nodo está en estado conectado y su última fecha de conexión.



En el siguiente pantallazo se observa el resultado de la detección del dispositivo IoT en la aplicación de gestión.

En el siguiente pantallazo se observa la información de los sensores asociados al dispositivo IoT que fue detectado.



En la consulta de la aplicación de gestión se muestra el siguiente mensaje:

```
agent 2: 192.168.2.114:161 [Disponible]
```

Detección de un dispositivo IoT “iniciado” parte del Gateway IoT

Al iniciar el Gateway IoT, este detecta los traps de tipo *heartbeat* enviados por el dispositivo IoT, lo cual se puede observar a través de la consola, así como a través de Wireshark.

```
---------------------------------------------
[SNMP] 2019/05/05 13:25:23
TRAP
Agente: 192.168.2.114/161
OID: 1.3.6.1.6.3.1.1.5.1
Type: heartbeat
---------------------------------------------
[SNMP] 2019/05/05 13:25:23
Agente: 192.168.2.114/161
Estado: [disponible]
---------------------------------------------
[SOAP]
Agente: 192.168.2.114/161
Resultado: Nodo actualizado
---------------------------------------------
```

No.	Time	Source	Destination	Protocol	Length	Info
2464	22.737680	192.168.2.114	192.168.2.102	SNMP	90	snmpV2-trap 1.3.6.1.6.3.1.1.5.50000.1

En la consulta de la aplicación de gestión se muestra el siguiente mensaje:

```
agent 2: 192.168.2.114:161 [No disponible]
```

Lo que ocurre posteriormente es igual al resultado obtenido cuando la detección se realiza mediante el trap *coldstart*.

RESULTADO DE LA PRUEBA DE DETERMINACIÓN DE UN DISPOSITIVO IOT COMO "NO DISPONIBLE"

Al apagar el dispositivo IoT, y luego de vencerse el temporizador de monitoreo en el Gateway IoT, este detecta los traps de tipo *heartbeat* enviados por el dispositivo IoT, lo cual se puede observar a través de la consola, así como a través de Wireshark.

Al apagar el dispositivo IoT, el Gateway IoT detecta su no disponibilidad con la expiración del temporizador de monitoreo, lo cual se ve reflejado en el mensaje que imprime en consola. A su vez, se indica que mediante el módulo de agente SOAP se ha realizado la actualización de esta información, en este caso hacia la aplicación de gestión.

```
---------------------------------------------
[SNMP] 2019/05/05 15:00:36
Agente: 192.168.2.114/161
Estado: [no disponible]
---------------------------------------------
 [SOAP]
Agente: 192.168.2.114/161
Resultado: Estado nodo actualizado
---------------------------------------------
```

A nivel de la aplicación de gestión, el cambio de estado del dispositivo IoT puede evidenciarse mediante el módulo de administración.

RESULTADO DE LA PRUEBA DE DETECCIÓN DE UN GATEWAY IOT POR PARTE DE LA APLICACIÓN DE GESTIÓN

Al inicializar el Gateway IoT, la aplicación de gestión detecta su disponibilidad mediante la recepción de un mensaje SOAP de heartbeat. Esto puede evidenciarse mediante el módulo de administración.

ID	NOMBRE	DIRECCIÓN IP	PUERTO	ESTADO	ÚLTIMA COCONEXIÓN
1	gestor 1	192.168.2.105	8888	CONECTADO	2019/05/05 02:40:14 PM
2	gestor 2	192.168.2.105	7777	DESCONECTADO	1969/12/31 07:00:00 PM

Adicionalmente, en la consola de administración se muestra el siguiente mensaje:

```
---------------------------------------------
[APLICACION]
Gateway (gestor 1): 192.168.2.102:8888 [disponible]
---------------------------------------------
[Cliente SOAP]
Gateway: 192.168.2.102:8888
Versión SNMP: 2c
Puerto trap SNMP: 10000
Puerto consola: 10570
**************************
Nodo: agent 1
Dirección: 192.168.2.105:161
Estado: false
**************************
Nodo: agent 2
Dirección: 192.168.2.114:161
Estado: true
```

RESULTADO DE LA PRUEBA DE DETERMINACIÓN DE UN GATEWAY IOT COMO "NO DISPONIBLE"

Al apagar el Gateway IoT, la aplicación de gestión detecta su no disponibilidad con la expiración del temporizador de monitoreo, lo cual se ve reflejado en su base de datos de gestión. Esto puede evidenciarse mediante el módulo de administración.

ID	NOMBRE	DIRECCIÓN IP	PUERTO	ESTADO	ÚLTIMA COCONEXIÓN
1	gestor 1	192.168.2.102	8888	DESCONECTADO	2019/05/05 03:10:02 PM
2	gestor 2	192.168.2.105	7777	DESCONECTADO	1969/12/31 07:00:00 PM

Adicionalmente, en la consola de administración se muestra el siguiente mensaje:

```
---------------------------------------------
[APLICACION]
Gateway (gestor 1): 192.168.2.102:8888 [no disponible]
---------------------------------------------
```

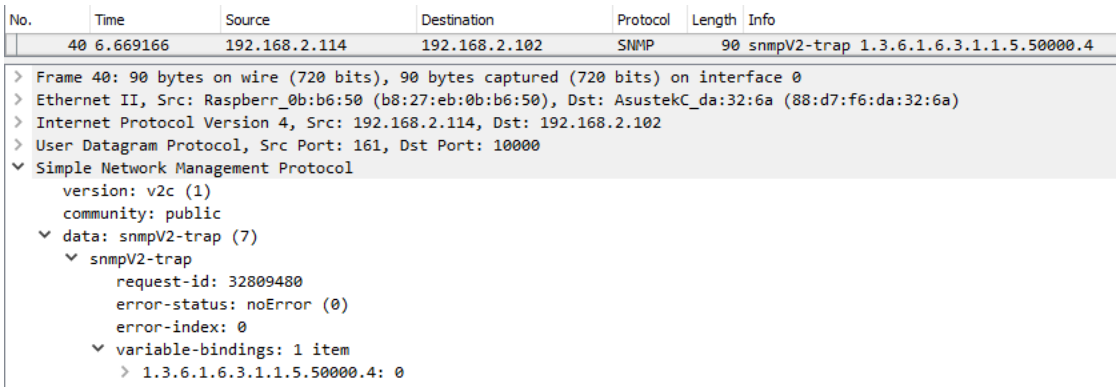
RESULTADO DE LA PRUEBA DE DETECCIÓN DE CAMBIO EN EL ESTADO DE UN SENSOR DEL DISPOSITIVO IOT

Para la realización de esa prueba, como se indicó en el protocolo, lo primero que se debe realizar es la desconexión de uno de los sensores del dispositivo IoT. En este caso, el seleccionada fue el sensor de ultrasonido.

Al realizar la desconexión del sensor, el dispositivo IoT detecta el cambio de estado el mismo y muestra en consola esta información.

```
---------------------------------------------
[Sensores] 2019/05/19 15:19:20
HC-SR04: inactivo (Operativo)
---------------------------------------------
```

Posteriormente, el dispositivo IoT envía un trap de tipo *sensorDownOper* (OID 1.3.6.1.6.3.1.1.5.50000.4), informando al Gateway IoT el cambio de estado del sensor.



Seguidamente, el Gateway IoT recibe el trap y actualiza su base de datos, al igual que envía la información de cambio de estado del sensor a la aplicación de gestión, lo cual observa en su consola.

```
---------------------------------------------
[SNMP] 2019/05/05 15:19:23
TRAP
Agente: 192.168.2.114/161
OID: 1.3.6.1.6.3.1.1.5.50000.4
Sensor: ULTRASONIDO
Estado: inactivo (Operativo)
---------------------------------------------
 [SOAP]
Agente: 192.168.2.114/161
Sensor: Sensor:HC-SR04 Driver:hc_sr04
Resultado: Estado sensor actualizado
---------------------------------------------
```

Finalmente, la aplicación de gestión recibe la información de cambio de estado del sensor y actualiza su base de datos de gestión, lo cual puede ser visualizado a través del módulo de administración.