_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE UPV INGENIEROS DE TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Desarrollo de una pasarela domestica para comunicaciones seguras en entornos IoT

César Rodríguez Ruiz

Tutor: Carlos Enrique Palau Salvador

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2015-16

Valencia, 1 de julio de 2016

A mis padres y mi hermano.

Resumen

El proyecto cubre todas las etapas de desarrollo para la implementación de una pasarela domestica de comunicaciones seguras en entornos IoT. El proyecto cubre la investigación y definición (arquitectura, mensajería, encriptación, etc.) para la posterior implementación de un caso de uso.

La función principal diseñada tiene el objetivo de asegurar la comunicación segura entre la pasarela y distintos elementos móviles de un entorno IoT. La funcionalidad diseñada cubre todo los procesos para asegurarse un entorno seguro de comunica iones:

- Instalación
- Envió y recepción de datos
- Migración de dispositivos IoT entre pasarelas.
- Desconexión
- Reconexión en caso de pérdida de señal

El diseño afronta las limitaciones ofrecidas por dispositivos IoT como son la baja capacidad computacional o la necesidad de implementar procesos de bajo gasto energético. Además, la funcionalidad desarrollada es compatible con distintos tipos de protocolos de comunicación como Wifi, Zigbee o Bluetooth.

El objetivo de principal de la funcionalidad desarrollada sería la configuración de diferentes casos de uso como la gestión domestica de e-health o domótica.

Resum

El projecte cobrix totes les etapes de desenrotllament per a la implementació d'una passarel·la domestica de comunicacions segures en entorns IoT. El projecte cobrix la investigació i definició (arquitectura, missatgeria, encriptació, etc.) per a la posterior implementació d'un cas d'ús.

La funció principal dissenyada té l'objectiu d'assegurar la comunicació segura entre la passarel·la i distints elements mòbils d'un entorn IoT. La funcionalitat dissenyada cobrix tot els processos per a assegurar-se un entorn segur de comunica ions:

- Instal·lació
- enviar i recepció de dades
- Migració de dispositius IoT entre passarel·les.
- Desconnexió
- Reconnexió en cas de pèrdua de senyal

El disseny afronta les limitacions oferides per dispositius IoT com són la baixa capacitat computacional o la necessitat d'implementar processos de baix gasto energètic. A més, la funcionalitat desenrotllada és compatible amb distints tipus de protocols de comunicació com Wifi, Zigbee o Bluetooth.

L'objectiu de principal de la funcionalitat desenrotllada seria la configuració de diferents casos d'ús com la gestió domestica d'e-health o domòtica."

Abstract

The project covers all the developments stages to implement a secure domestic communications gateway for IoT environments. The project covers the research and definition (architecture, messaging, encryption, etc.) for the subsequent implementation of a use case.

The main feature designed aims to ensure secure communication between the gateway and mobile devices in IoT environment. The functionality covers all processes and stages of an enviroment to ensure a safe communications:

- Installation
- Send and receive data
- Migration IoT devices between gateways.
- Disconnection
- Reconnection in case of signal loss

The design addresses the limitations offered by IoT devices such as low computational capacity or the need to implement processes of low energy cost. In addition, the functionality developed is compatible with different types of communication protocols such as WiFi, Zigbee or Bluetooth.

The main objective of the functionality developed would be setting different use cases such as domestic e-health management or home automation.

Índice

Índice de Figuras y Tablas

Figuras:

Tablas:

Capítulo 1. Introducción

1.1 Motivación

El nacimiento de las redes de comunicaciones sobre medios como internet y sus protocolos asociados se produjo sin contemplar la idea usuarios maliciosos. No se contempló la posibilidad de que la información transmitida por este tipo de medios iba a ser susceptible de ser robada e utilizada con fines maliciosos. Por esta razón la seguridad tomo una importancia secundaria a la hora de desarrollar las herramientas para este tipo de comunicaciones.

Con el paso del tiempo y el crecimiento de estas redes los términos *Privacy, Security and Trust* no ha hecho sino crecer en importancia. Hoy en día, a través de las redes de comunicaciones se información tremendamente sensible y que debe ser totalmente confidencial. Sectores como la banca han invertido en el desarrollo de herramientas para garantizar la seguridad de sus comunicaciones. Hoy en día, el sector de la seguridad se ha convertido en uno de los pilares principales de desarrollo en el sector de la comunicaciones para proteger la información de millones de usuarios de la redes.

Los sectores de IoT están en pleno desarrollo y se prevé un crecimiento enorme en el sector. Para 2020 se prevé que hayan desplegados 50 mil millones de dispositivos y que el sector generara un negocio de entorno a los 19.000 millones de dólares en los próximos 10 años.

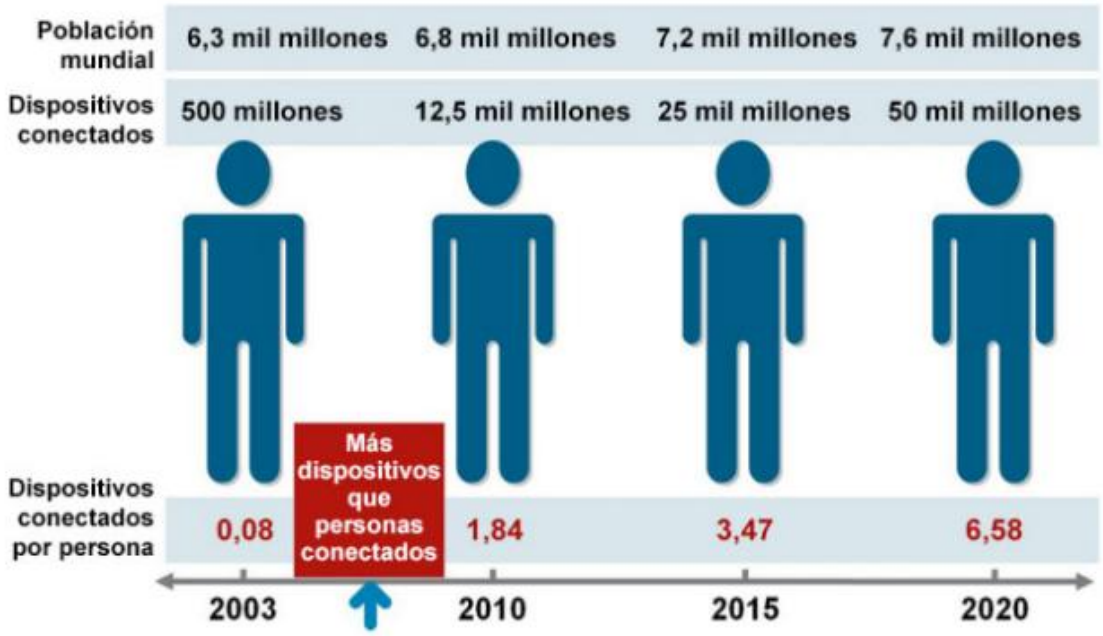

Figura 1. Crecimiento de dispositivos IoT

En vista de la creciente criticidad de los datos en este tipo de entornos y la falta de estandarización en las comunica iones del sector IoT, este proyecto pretende llenar ese hueco y combinar las herramientas existentes para conseguir proveer de una seguridad solida a un entorno de tamaño medio o domestico, cubriendo las necesidades y limitaciones de los dispositivos envueltos en los entornos IoT. La funcionalidad desarrollada pretende garantizar una comunicación segura y solida frente a ataques de usuarios maliciosos con el objetivo de forma parte de sistemas domóticos o e-health, este último sector altamente crítico debido a la información que transmitirían dispositivos: medidas de azúcar en sangre, ritmos cardiacos, etc. Cualquier intruso que pudiera alterar dicha información podría provocar un serio problema médico al usuario, por tanto es necesaria la implementación de un entorno totalmente solido y seguro para la captación y trasferencia de datos a la red.

Una vez definida y desarrollada la funcionalidad de implementará un caso de uso (pasarela en entorno domestico) para comprobar la validez y solidez de dicho desarrollo.

1.2 Estructura de documento

Para guiar al lector en la lectura de este Trabajo de Fin de Grado, a continuación, se describirá la estructura del documento:

Introducción

Se introduce el proyecto resumiendo cuál es la motivación del proyecto, qué problema se desea resolver y una breve descripción sobre la estructura del documento en cuestión.

Gestión y Organización del proyecto

En este capítulo se definen las etapas que se siguen durante el desarrollo de software y, por tanto, este proyecto.

Estado de la cuestión

En este punto se realizar la investigación y en análisis actual sobre la situación de la cuestión que se pretende analizar para consolidar la base teórica sobre la cual se sustentara el desarrollo del proyecto y el software.

Desarrollo del software

Se desarrollaran y explicaran las etapas de objetivos, análisis, definición y diseño para la posterior implementación de la funcionalidad y su caso de uso.

Caso de Uso

Análisis y explicación del caso de uso contemplado en la fase de desarrollo y ejemplo práctico de la situación teórica.

Conclusiones y líneas futuras

Se detallan las conclusiones tras la finalización del proyecto y se establecen una serie de líneas futuras de desarrollo para mejoras futuras del software.

Capítulo 2. Gestión y Organización del Proyecto

En este capítulo se va a detallar, qué es la ingeniería del software, qué metodologías se pueden seguir en un proyecto de este tipo, qué ciclos de vida sigue el software, así como la distribución en tareas que se ha seguido para el desarrollo de la aplicación de este proyecto.

Durante el desarrollo de software no es suficiente con la ejecución de la fase de implementación. Como recomienda la guía de ingeniería del software del Laboratorio Nacional de Calidad del Software de INTECO [1], en un proyecto de desarrollo se deben seguir una serie de metodologías o procedimientos para evitar problemas pasado el lanzamiento del mismo. El grupo de trabajo que forma parte del desarrollo debe conocer que ciclo de vida sigue y en que fase de la producción se encuentra el software en cada momento para ser capaces de realizar correctamente y eficientemente su trabajo.

2.1 ¿Qué es IoT?

Es un concepto que viene asociado a la interconexión digital de dispositivos cotidianos. El nombre le viene gracias a Kevin Ashton en 1999, se le conoce por desarrollar en el departamento Auto-ID Labs del MIT el estándar global de comunicación de RFID. Existen muchas definiciones para este concepto pero no existe una forma estandarizada de explicar qué es.

Figura 2. Diagrama Comercial IoT

A modo de resumen el Internet de las Cosas es la comunicación de una cantidad elevada de dispositivos de bajo consumo que mediante su interacción en un entorno limitado permite la automatización y control del mismo.

2.2 ¿Qué es *SOFTWARE*?

La RAE [1] define software como "Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora". En cambio, en el proceso de desarrollo de software intervienen más factores como los datos y la documentación. Por tanto se entiende que el software es un elemento lógico que se DESARROLLA no se fabrica.

Los programas e instrucciones proporcionan la funcionalidad y el rendimiento de un software cuando se ejecuta. Están escritos usando lenguajes de programación que los ordenadores pueden leer y ejecutar. Los datos hacen referencia a todas las estructuras de datos e información que permiten manejar y probar un programa. En último lugar y no por ello menos importante, se encuentran los documentos, que abarcan toda la información que describe el uso y operación de un programa para facilitar el manejo y la posible modificación del mismo.

2.3 Etapas del desarrollo de software

Vista la definición de software de puede intuir que el proceso de desarrollo de software no es sencillo. Se trata de un proceso compuesto por varias etapas que se deben atravesar para garantizar la calidad del producto desarrollado y la fiabilidad y robustez del mismo. EL orden que se sugiere seguir a la hora del desarrollo, a grandes rasgos, es el siguiente:

Análisis de requisitos

La primera etapa en el proceso del desarrollo de software es el análisis de requisitos que además se podría considerar la más importante de todas. Pues si no se definen bien los requisitos, difícilmente se obtendrá un software adaptado a las necesidades del cliente. El resultado al final de esta etapa es un documento conocido como: Especificación de requisitos. El estándar que normaliza la creación de las especificaciones de requisitos software es el *830-1998* de *IEEE*.

Especificación

Esta fase del proceso corresponde con la escritura detallada por parte del desarrollador del software de forma rigurosa. Es decir, describe todas las características importantes del producto, sus características deseables y el formato adecuado de representación.

Diseño

En la fase de diseño, se determina cómo funcionará el software de una forma general. Sin entrar en detalles, se deben añadir las consideraciones acerca de la implementación, el hardware que se utilizará, la estructura que seguirá el programa, etc. Además se definirán los casos de uso que abarquen todas las funciones que realizará el software.

Programación

La fase más evidente en el desarrollo de software. En esta etapa se escribe en un lenguaje de programación el código que tendrá como finalidad cumplir con los requisitos, especificaciones y diseño del producto. Esta fase, no es necesariamente la que requiere de mayor dedicación de tiempo. Si se han realizado de forma correcta las fases anteriores, el proceso de programación se facilita enormemente.

Prueba

También conocida como fase de test. En la fase de pruebas se realizan las comprobaciones para analizar si el software cumple con los requisitos especificados. Los test abarcan desde pruebas unitarias de los distintos módulos y su funcionalidad a pruebas de integración del sistema completo. Para realizar correctamente esta etapa, la debe realizar una persona distinta a la que desarrollo el software.

Mantenimiento

La última fase es la de mantenimiento. Un software lanzado sin mantenimiento posterior, puede quedar inútil en poco tiempo. Dado que siempre quedan fallos que no se han corregido o imprevistos, o que se añaden nuevos requisitos después de haber lanzado el software, es necesaria una etapa de mantenimiento.

Existen 4 tipos de mantenimiento. El mantenimiento perfectivo se realiza para mejorar la calidad interna del software, el mantenimiento evolutivo se realiza para adaptarse al cambio de las necesidades del cliente o usuario, el mantenimiento adaptativo se realiza para adaptarse a los cambios de hardware y software con los que trabaja el producto y por último, el mantenimiento correctivo se encarga de la subsanación de errores.

2.4 Metodología y Ciclo de vida de desarrollo de software

Visto el apartado anterior, para el correcto desarrollo de software de debe seguir una serie de normas, reglas y etapas definidas en una metodologías de desarrollo de software. "Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo."[1] Siguiendo una metodología, se consigue la ejecución exitosa del proceso de desarrollo y cada una de las etapas que intervienen.

El uso de una metodología implica una correcta estructuración del trabajo y ser eficiente al ejercer las distintas tareas que cada etapa implica. Dentro de una metodología de definen una serie de fases consecutivas compuestas por tareas panificables que conforman un ciclo de vida. Dependiendo del tipo de ciclo de vida, la sucesión de etapas se podrá ampliar mediante la realimentación entre fases para que una misma fase se pueda ejecutar más de una vez durante la ejecución de un proyecto.

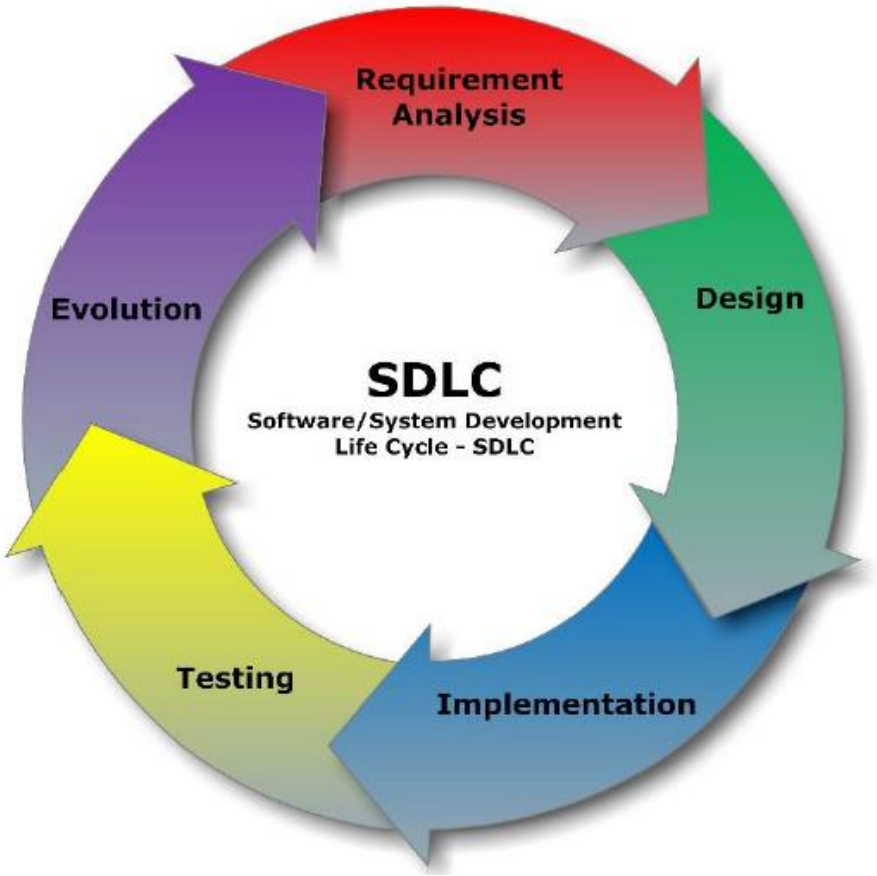

Figura 3. Ciclo de vida del desarrollo de software

2.4.1 Modelos de ciclo de vida

Existen varios modelos de ciclos de vida para que se aplican al desarrollo de software: modelo Royce o Cascada, en V, iterativo, incremental, etc.… Para este proyecto se ha utilizado el modelo incremental, que es una mezcla entre el modelo en cascada e iterativo explicados a continuación:

2.4.1.1 Modelo de Royce o Cascada

Fue el primer modelo en aparecer y el más sencillo compuesto por las siguientes fases: especificación de requisitos, diseño, implementación, integración, pruebas, instalación y mantenimiento. Cada fase a de esperar a la finalización de la anterior para su inicio. Simple y fácil de gestionar, es ventajoso para proyectos pequeños en los que las necesidades y requisitos quedan bien definidos desde el inicio y no varían durante el desarrollo de las sucesivas etapas. En cambio, si los requisitos y necesidades cambian contantemente, el proyecto queda bloqueado ya que no se permite la movilidad entre capas.

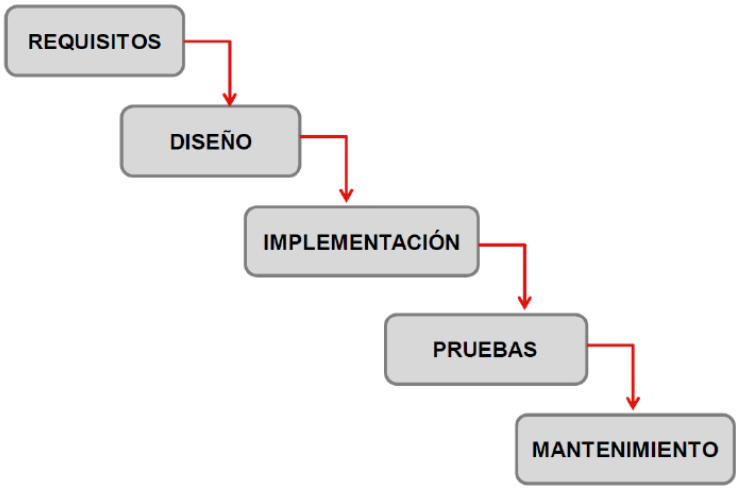

Figura 4. Modelo de ciclo de vida en cascada

2.4.1.2 Modelo iterativo

Deriva del modelo en cascada pero trata de corregir el problema de la redefinición de requisitos. Se base en la iteración de varios ciclos de vida en cascada en la que cada final de ciclo implica un entregable nuevo del producto con todos los cambios necesarios.

Una ventaja fundamental es que para iniciar el proyecto no es necesario que estén definidos todos los requisitos ya que en cada iteración se pueden ir modificando y adaptando a las necesidades del usuario.

La desventaja de este modelo radica en que al ser posible la iniciación del proyecto sin tener todos los requisitos establecidos, se definirá una estructura determinada en la fase de diseño del software, que posiblemente causará problemas al definir por completo los requisitos ya que requerirá cambios en la arquitectura del software.

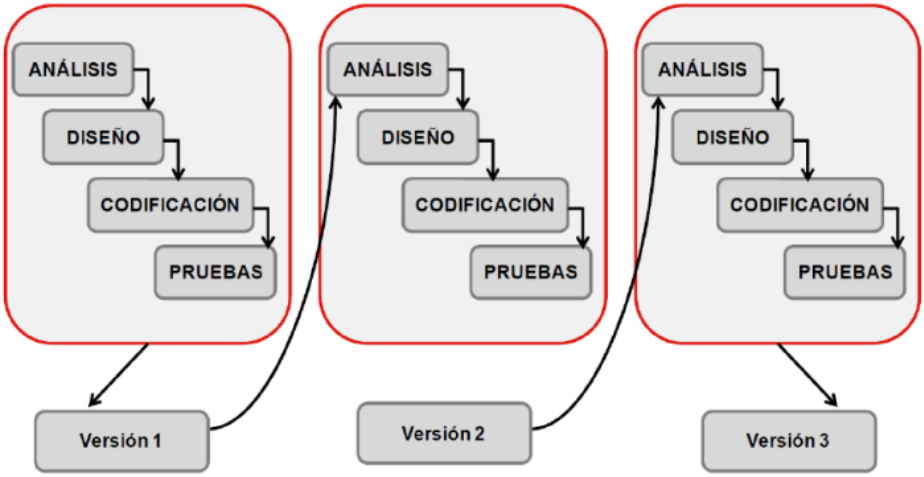

Figura 5. Modelo de ciclo de vida iterativo

2.4.1.3 Modelo Incremental

Como se ha dicho antes, hereda características de los 2 anteriores. En esencia, el modelo es muy parecido al iterativo. La diferencia fundamental se encuentra en que la primera versión del modelo incremental es un producto software esencial que cumple con los requisitos más básicos establecidos por el cliente. Conforme se vayan realizando más iteraciones durante la ejecución del proyecto, las siguientes versiones del software serán más completas conservando las estructuras establecida desde el primer momento.

Con este modelo es posible generar software operativo de forma rápida en las primeras fases del proyecto además de que se trata de un modelo más flexible que los anteriores. Es más fácil realizar pruebas en cada iteración, lo que es una gran ventaja y previene futuros errores.

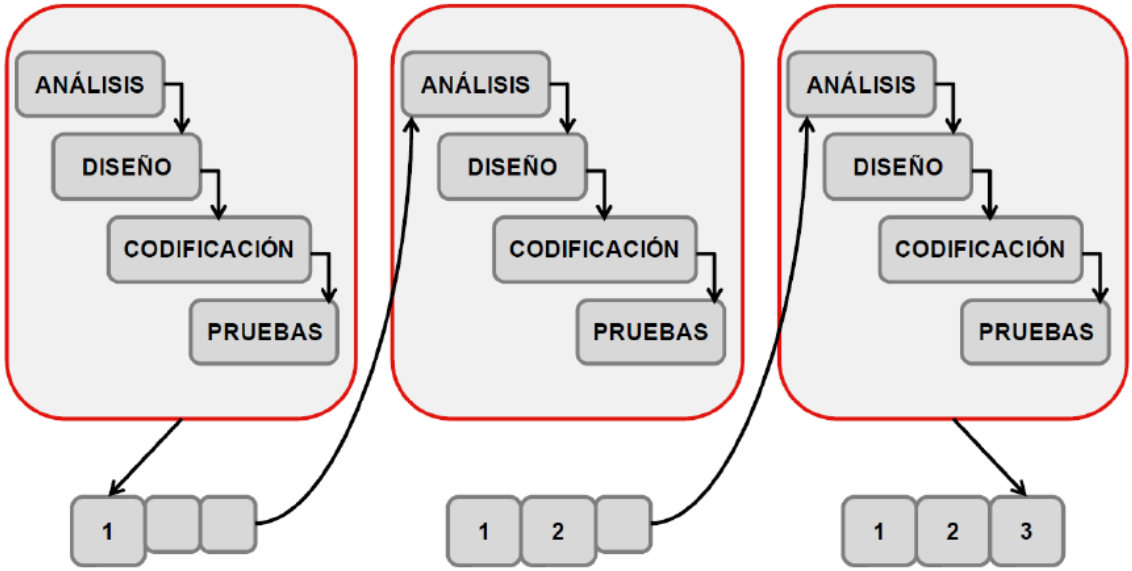

Figura 6. Modelo de ciclo de vida incremental

El proyecto se desarrollara utilizando este modelo. Se definirá una funcionalidad básica a la que se irá atribuyen mayor complejidad y en cada iteración se probará la funcionalidad desarrollada. Por tanto de podrán desarrollar y probar por separado los distintos estados del proyecto desarrollado:

- Instalación
- Envió y recepción de datos
- Migración de dispositivos IoT entre pasarelas.
- Desconexión
- Reconexión en caso de pérdida de señal

Capítulo 3. Estado de la Cuestión

En este capítulo se pretende investigar y analizar el estado actual del entorno que rodea al proyecto, en nuestro caso las comunicaciones y herramientas de seguridad en entornos IoT.

Lo primero que se denota es que NO hay un sistema claro se securización para los sistemas de IoT. Las soluciones son, básicamente, aplicaciones de métodos de securización conocidos aplicados y adaptados a esta plataforma debido a las limitaciones de los equipos implicados en ella.

Los equipos que forma el perímetro de los sistemas de IoT son normalmente pequeños, baratos y sin ningún tipo de seguridad física. Son equipos muy limitados en memoria y capacidad computacional, lo que provoca que no sean compatibles con algoritmos de seguridad cambiantes y complejos, pues este tipo de securización requiere alta capacidad de procesamiento. Además que el numerosísimo numero de dispositivos hace muy complicada la gestión centralizada de los mismos o la monitorización de la red. Debido a esta razón, los bajos recursos de los dispositivos, a priori queda ya descartada cualquier solución en la que intervenga el uso de certificados pues requiere una serie de privilegios que los muchos dispositivos simple son tendrán. Además de descartar el uso de certificados, la utilización de PKI también es desaconsejable debido a los requerimientos de hardware que ello requeriría. En vista de las necesidades (limitaciones), a priori, el entorno estaría basado en el uso de clave simétrica para el cifrado de datos y en algún mecanismo alternativo para la autenticación de dispositivos.

Otro punto crítico se encuentra en el consumo de la información captada por los sistemas. Se requerirán sistemas de acceso equitativo y abierto a los datos y al mismo tiempo se deberá conservar la privacidad y la exclusividad entre los consumidores. Debemos establecer los controles de identidad adecuados y construir relaciones de confianza entre entidades para compartir la información de forma correcta.

Por tanto se requiere afrontar:

- Autenticación en múltiples dispositivos de una red.
- Disponibilidad total y 24/7 de datos para múltiples usuarios simultáneamente garantizando confidencialidad.
- Proporcionar una sólida autenticación y protección de datos.
- Permitir y soportar la evolución de los sistemas de seguridad de cara a los riesgos futuros y desconocidos.

En vista de esta necesidades, a continuación analizaremos una serie de protocolos utilizados en entornos IoT además de una serie de herramientas de securizacion para decidir y diseñar la herramientas más adecuadas para nuestra entorno: protocolos, claves, dispositivos, etc.

3.1 Protocolos IoT

3.1.1 6LowPAN - Thread

6LoWPAN (IPv6 de baja potencia inalámbrica de red de área personal). En lugar de ser un protocolo de capa de aplicación como Bluetooth o ZigBee, 6LoWPAN es un protocolo de red que define los mecanismos de encapsulación y compresión de la cabecera. El estándar tiene libertad en el uso de la banda de frecuencia y en la capa física y también se puede utilizar a través de múltiples plataformas de comunicaciones, incluyendo Ethernet, Wi-Fi, 802.15.4 y sub-1GHz ISM. Un atributo clave es la pila de IPv6 (Protocolo de Internet versión 6), sucesor de IPv4 ofrece aproximadamente 5 x 10^28 para cada persona en el mundo, permitiendo a cualquier objeto o dispositivo integrado en el mundo tener su propia dirección IP única y conectarse a Internet.

Especialmente diseñado para la domotica o la automatización de edificios, por ejemplo, IPv6 proporciona un mecanismo de transporte básico para producir sistemas de control complejos y para comunicarse con los dispositivos de una manera rentable a través de una red inalámbrica de bajo consumo.

Diseñado para enviar paquetes de IPv6 en las redes basadas en IEEE 802.15.4 y compatible con TCP, UDP, HTTP, COAP, MQTT y websockets. 6LoWPAN construye redes malladas robustas, escalables e auto-reparables.

Thread:

Basado en estándares abiertos y protocolos IPv6/6LoWPAN. Utiliza cifrado de clase bancario para solventar problemas de seguridad que existen en otros protocolos inalámbricos. Utiliza de AES para la encriptación.

3.1.2 ZigBee

Es un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal. Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías, características básicas de sistemas IoT.

Proporciona tanto mecanismos de control de acceso de los dispositivos a la red (autenticación) como de cifrado (utilizando criptografía de clave simétrica) así como de integridad, asegurando que las tramas transmitidas no han sufrido manipulación con comprobaciones de integridad de mensaje (MIC). Soporta cifrado por medio de AES-128 bits.

3.1.3 CoAP

Constrained Application Protocol - Protocolo de software destinado a ser utilizados en dispositivos electrónicos muy simples que les permite comunicarse de forma interactiva a través de Internet. Está dirigido especialmente a los pequeños sensores de baja potencia, interruptores, válvulas y componentes similares que necesitan ser controlados o supervisados de forma remota, a través de las redes de Internet.

Es un protocolo de capa de aplicación que está destinada para su uso en dispositivos de IoT o M2M, traduce fácilmente a HTTP para la integración simplificada con la web. Compatible con la multidifusión, es muy simple y requiere de un bajo coste de energía para la transmisión de catos. COAP se puede ejecutar en la mayoría de los dispositivos que admiten UDP o un análogo de la UDP. La transmisión se realiza mediante DTLS para evitar escuchas no deseadas (eavesdropping), accesos no permitidos, o modificación de mensajes. DTLS basado en TLS, equivalente a claves de 3072-bits RSA. Soporta cifrado por medio de AES-128 bits.

3.1.4 MQTT

Protocolo diseñado para el intercambio de mensajes extremadamente ligero. Diseñado para dispositivos limitados y bajo ancho de banda y latencia alta o redes no confiables. Los principios de diseño son para minimizar el ancho de banda de la red y los requisitos de recursos del dispositivo a la vez que el intento de asegurar la fiabilidad y cierto grado de garantía de entrega.

A diferencia de otros protocolos de para la transmisión de datos, este protocola dota de cierta seguridad a la comunicación (mínima, basado en un ID). Se puede identificar un cliente MQTT por su identificador de cliente, su ID de usuario o su certificado digital público. Aun así para securizar el intercambio de información, el protocolo funciona sobre SSL para realizar funciones de autenticación y autorización. Para mantener la ligereza en el intercambio de mensajería, la seguridad no se ha mantenido en segundo plano pues añade carga al intercambio de mensajes y la construcción de los mismos.

3.2 Herramientas de Seguridad

3.2.1 TLS/SSL

Por regla general es uso de TLS/SSL es la solución para garantizar seguridad en cualquier comunicación. Por esta razón analizaremos sus características y aplicaciones a nuestro entorno.

Aplicaciones cliente-servidor utilizan el protocolo TLS para comunicarse a través de una red de manera segura para prevenir el espionaje y la manipulación de datos.

Dado que los distintos protocolos de transporte pueden funcionar ya sea con o sin TLS (o SSL), es necesario que el cliente indique al servidor el establecimiento de una conexión TLS. Hay dos maneras principales de conseguir esto. Una opción es usar un número de puerto diferente para las conexiones TLS seguras que las inseguras o que el cliente utilice un mecanismo específico del protocolo para solicitar que el servidor cambie la conexión a TLS (STARTTLS para protocolos de mail).

Una vez que el cliente y el servidor han accedido deben utilizar TLS, negocian una conexión, lo que se llama *handshake.* Durante este apretón de manos, el cliente y el servidor se ponen de acuerdo en los parámetros utilizados para establecer la conexión de manera segura:

i) El apretón de manos comienza cuando un cliente se conecta a un servidor habilitado para el uso de TLS. El cliente presenta una lista de mecanismos de cifrado soportados (sistemas de cifrado y funciones hash).
ii) De esta lista, el servidor escoge una función de cifrado y hash que también soporta y notifica al cliente la decisión.
iii) El servidor, por lo general, envía de vuelta su identificación en forma de certificado digital. El certificado contiene el nombre del servidor, la autoridad de certificación de confianza (CA) y clave de cifrado pública del servidor.
iv) El cliente confirma la validez del certificado antes de proceder.
v) Para generar las claves de sesión utilizados para la conexión segura, el cliente puede:

    a) Encriptar un número aleatorio con la clave pública del servidor y envía el resultado al servidor (sólo el servidor debe ser capaz de descifrar dicho numero con su clave privada); ambas partes a continuación, utilizan el número aleatorio para generar una clave de sesión única para el cifrado y descifrado de datos posterior durante la sesión y transmisión de datos.

b) Utilizar Diffie-Hellman para generar de forma segura una clave de sesión aleatoria y única para el cifrado y descifrado que tiene la propiedad adicional de confidencialidad directa: si la clave privada del servidor se da a conocer en el futuro, no se puede utilizar para descifrar la sesión actual, incluso si la sesión es interceptada y guardada por un tercero.

Con esto concluye *handshckey* comienza la conexión segura, que se cifran y descifran con la clave de sesión hasta que la conexión se cierra. Si alguno de los pasos anteriores fallan, el apretón de manos TLS falla, y la conexión no se crea.

TLS garantiza la integridad de los datos por medio del uso de MAC. Los valores MAC se calculan mediante la aplicación de una función hash criptográfica con clave secreta *K*, que sólo conocen el remitente y destinatario, pero no los atacantes. Claves posteriormente calculadas o conocidas en el proceso de *handshake*.

Se observa que un punto tremendamente fuerte de TLS es de uso de claves simétricas solo conocidas por emisor y receptor. Si se evita toda la gestión de generación de claves, se resta de complejidad computacional al proceso. En cambio habrá que buscar un modelo eficiente y seguro de gestión de dicha claves.

3.2.2 Kerberos

El objetivo de estudiar este protocolo es analizar la figura del servidor de autenticación. Adelantándonos al proceso de definición, como se ha dicho con anterioridad, el uso de PKI no es adecuado para dispositivos con baja capacidad computacional. Por tanto cuando el tamaño de la red en la que nuestros dispositivos intercambian información sea muy grande, deberá haber algún dispositivo que autentique dichos dispositivos y distribuya sus claves. Con este objetivo, se analiza este componente, el SA.

El propósito del servidor de autenticación es el de definir los usuarios autorizados a acceder al sistema y tener posibilidades de acceder al TGS. Kerberos basa su funcionamiento en la existencia de un secreto compartido entre el cliente y el servidor de autenticación.

Kerberos realmente implementa tres protocolos diferentes de forma simultánea

-Protocolo de autenticación

-Protocolo de obtención de un ticket (TGT)

- Protocolo Cliente/Servidor

Vista la función de dicho componente en una red basada en Kerberos, a la hora de diseñar redes de gran tamaño con múltiples dispositivos, el diseño tendrá un componen parecido a la figura del SA en kerberos.

Capítulo 4. Desarrollo del software - Diseño

Una vez realizado el análisis de la cuestión, es hora de llevar a cabo la definición de los requisitos y la arquitectura..

4.1 Punto de Partida

Analizando la investigación realizada sobre los distintos protocolos del entorno en el que se desarrolla el proyecto se observa que las capacidad de securizar la comunicaciones de dichos protocolos es muy pequeña. En la mayoría de los casos observados de apela al uso de TLS para garantizar la seguridad de los comunicaciones. Como se ha dicho con anterioridad, una de las fortalezas de TLS es la utilización de claves simétricas de sesión para el cifrado de datos. Dichas claves de cifrado provienen de unos algoritmos de negociación como Diffie-Hellman o RSA. Para dispositivos con capacidades computacionales suficientes, estas negociaciones son factibles y posibles pero, en el entorno que nos encontramos, dichas capacidades son tremendamente limitadas. Debido a la situación que se debe manejar, se plantea la cuestión de proporcionar un alto grado de seguridad, equivalente a TLS, sin procesos de negociaciones de claves. Mediante un proceso de despliegue e instalación, dichas claves van a ser desplegadas por el usuario administrador de la red y almacenadas en la Pasarela Maestra o Servidor de Autenticación que explicaremos en el apartado 4.2 Arquitectura. Dependiendo del tamaño de nuestra red, la función de autenticación será realizado por un pasarela determina o un servidor de autenticación, externo a esta.

Con la distribución de dichas claves, solamente conocidas por el administrador y distribuidas por un canal secundario seguro, ningún paquete de nuestra red viajara en claro. Los datos de identidad y autenticación serán prefijados por el administrador durante el proceso de instalación para la primera conexión de dispositivos a la red y a partir de ahí, en cada migración entre pasarelas, estos parámetros serán reconfigurados por las pasarelas.

En vista de que los distintos protocolos de entornos IoT soportan AES-128 bits, será este el algoritmo de encriptación elegido para nuestra red ya que está catalogado como totalmente seguro y hoy en día no hay herramientas para romperlo por métodos de fuerza bruta.

Basados en el análisis de los protocolos y demás herramientas de seguridad se ha definido los siguiente componentes de nuestro proyecto

4.2 Arquitectura

En este proyecto se van contemplar 2 posibles arquitecturas dependiendo del tamaño de nuestra red: entorno domestico (pequeño- un hogar) o entorno local (medio – hospital, universidad, etc.)

4.2.1 Arquitectura de entorno domestico.

Destinado a un emplazamiento domestico esta distribución está diseñado en la mayor simplicidad posible para la gestión de nuestra red. Para este tipo de red se definen 3 tipos de

dispositivos: Pasarela Maestra, pasarela estándar y dispositivo IoT. En este tipo de entorno se espera encontrar un pasarela maestra y 2, como mucho 3, pasarelas estándar más.

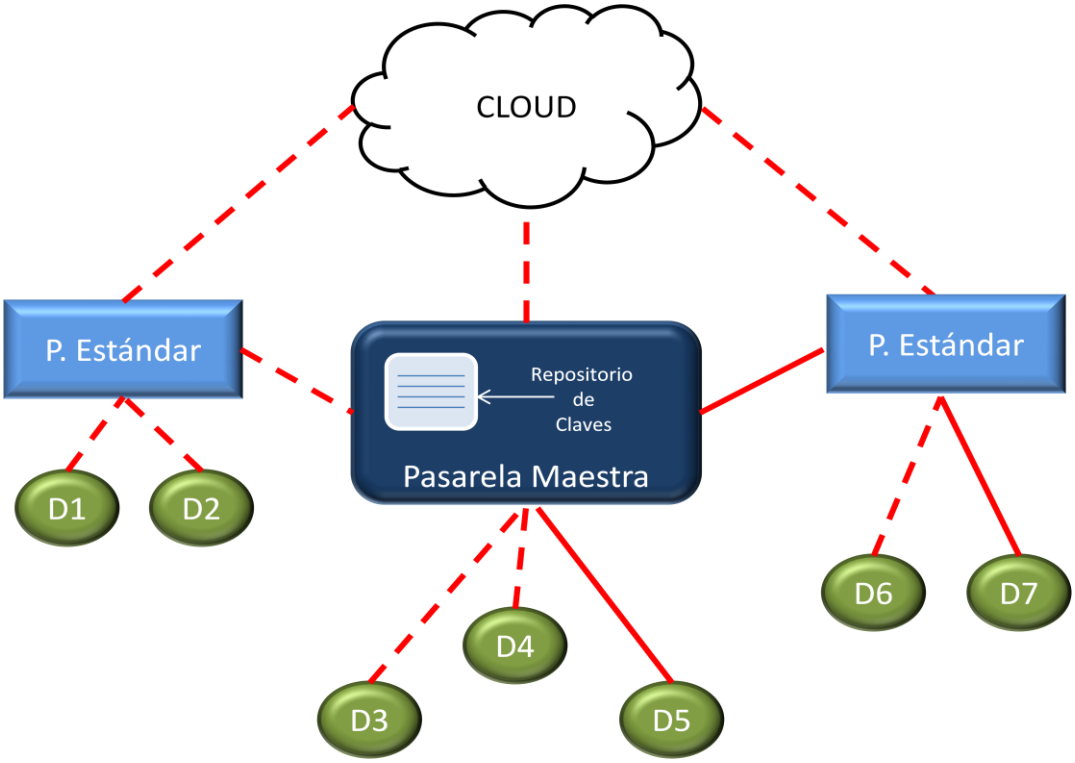

Figura 7. Arquitectura de entorno doméstico

i) <u>Pasarela Maestra (PM)</u>

Al ser una red un tamaño muy pequeño, equivalente a un hogar cualquiera, no es necesaria la figura del servidor de autenticación. El punto central de gestión seria la Pasarela Maestra. Por medio de un usuario y contraseña el usuario administrador puede acceder a dicho dispositivo para añadir, borrar o alterar cualquier dispositivo o pasarela estándar de nuestra red. Este dispositivo alberga una base de datos a la que solo se puede acceder mediante usuario y contraseña en la que se almacenan las distintas claves de los elementos de nuestra red local. Este dispositivo es el encargado de distribuir y administrar los datos por la red.

Además de la función de administración de nuestra red, este dispositivo capta información de la red que proviene de los dispositivos y los envía al repositorio de datos en la nuble o servidor de base de datos que marque el usuario administrador.

ii) <u>Pasarela Estándar (PE)</u>

Estos dispositivos solo realizan la función de transmisión de datos entre dispositivos y la nuble o servidor de base de datos. Albergan los datos referentes a los dispositivos que están transmitiendo y al resto de pasarelas de nuestro hogar para las posibles migraciones hacia cualquier pasarela de la red.

La información guardada referente a dispositivos es:

- IP asignada
- Identificador Único
- Numero de secuencia
- Clave de cifrado
- Marca de tiempo de la última recepción de datos.

- Status: Activo / No Activo

La información guardada referente a pasarelas vecinas es:

- IP asignada
- MAC
- Clave de cifrado
- Puerto de Escucha

Se guarda temporalmente información sobre los dispositivos que han estado conectados a ella durante un periodo de tiempo para, en caso de reconexión por la pérdida de la señal, no tener que consultar sus datos y minimizar la mensajería de la red. Durante las tareas de mantenimiento se gestiona este repositorio temporal de dispositivos.

iii) Dispositivo

Son los elementos de nuestra red encargados de la captación de los datos. Estos dispositivos solamente albergan su propia clave además de la clave de cifrado del dispositivo al cual están transmitiendo.

El proyecto contempla la posibilidad de soportar dispositivos que manejan datos de criticidad variable. No se le puede dar el mismo tratamiento a un sensor de temperatura que a un dispositivo de e-health. Con el objetivo de reutilizar el identificador único asignado al dispositivo, este también se utiliza para definir la criticidad de los datos manejados. De esta manera se soporta el uso no de ACKs en el envío de datos o distintas alertas durante las tareas de mantenimiento.

En caso de quiere firmar los datos enviado además deberá albergar su clave privada de firmado.

4.2.2 Arquitectura de entorno local.

Destinado a un emplazamiento local, por ejemplo un hospital o centro de mayores, esta distribución está diseñada para ser más versátil y hacer posible la gestión de nuestra red de forma eficiente pues el emplazamiento sea mayor geográficamente. Al contrario que la red domestica, por medio del servidor de autenticación, el número de pasarelas de nuestra red local puede aumentar considerablemente, pues tenemos un dispositivo diseñado expresamente para este objetivo, se extrae la función de autenticación de la pasarela maestra. Además, en caso de tener una red de un tamaño considerable, por medio de servidores proxy, la Para este tipo de red se definen 3 tipos de dispositivos: Servidor de autenticación e identidad, pasarelas estándar y dispositivo IoT.

Básicamente el funcionamiento de los entornos es el mismo redireccionando las consultas de autenticación de dispositivos en vez de a la Pasarela Maestra al SA. Además en caso de que nuestra red crezca, por medio de servidores proxy conectados a nuestro SA, se podría balancear el tráfico para resolver este tipo de consultas. Se debería dimensionar estos dispositivos con mayores recursos para poder generar claves para realizar encriptaciones robustas y garantizar la distribución segura de claves.

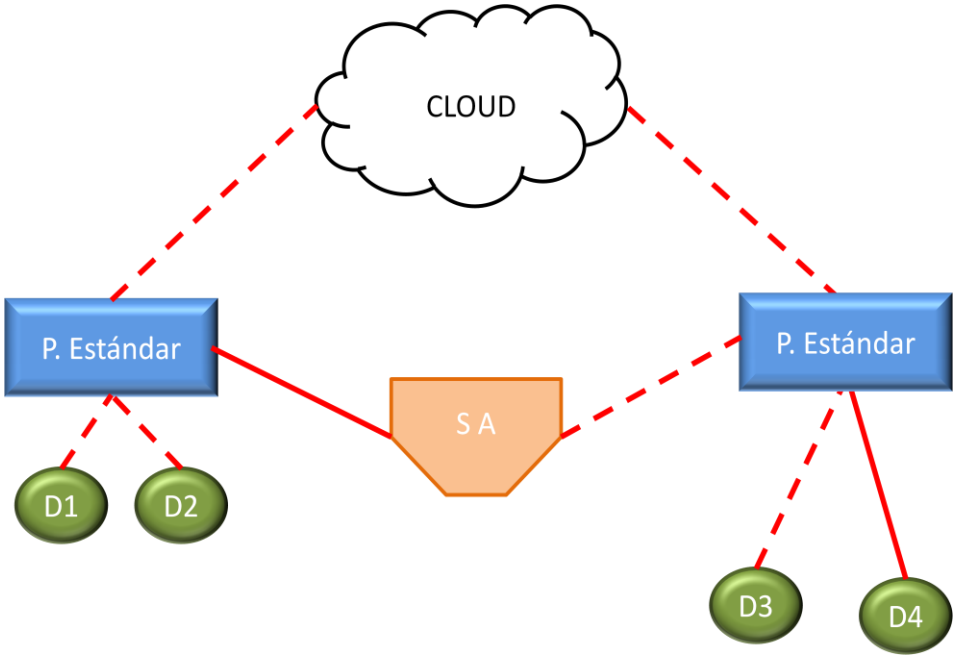

Figura 8. Arquitectura de entorno local

Siendo las pasarelas estándar y dispositivos idénticos a los ya definidos en el entorno domestico, solo queda definir la función del servidor de autenticación.

i) **<u>Servidor de Autenticación (SA)</u>**

En este caso, la función de administración de la red no está ligada a un dispositivo pues sería ineficiente debido al mayor número de dispositivos. Es necesaria la figura de un servidor de autenticación para el desempeño de la función de identificación de dispositivos y distribución de claves.

En la etapa de instalación, los elementos de la red (Dispositivos y PE) son registrados en el servidor de autenticación, y esta almacena su clave y sus datos de identidad. A la hora de

conectarse un dispositivo a una pasarela, este consultara la identidad de dicho dispositivo y su validez en el servidor de autenticación. En el caso de migración de dispositivos entre pasarelas, en caso de que las pasarelas "no se conozcan", deberán de consultar su identidad y datos también en el SA.

4.3 Herramientas de Seguridad

Una vez definida la arquitectura de nuestros entonos en base a su tamaño y necesidades, es hora de definir que herramientas de seguridad se han decidido utilizar para securizar nuestro entorno. El punto clave de nuestro sistema es la NO existencia de tráfico en plano por nuestra red. Todo el tráfico de nuestra red está cifrado. En la etapa de instalación se definen las claves de los dispositivos y por medio de un canal alternativo (el usuario es quien define la claves antes de conectar los dispositivos a la red) las claves no viajen por la red. Se trata de realizar un proceso de instalación que puede resultar costoso pero a la par garantiza una alta seguridad en el entorno. Además, el tiempo dedicado a la instalación del sistema da robustez e independencia para operar 24/7 de forma autónoma.

4.3.1 Encriptación

En vista de las necesidades de universalidad del sistema y vista la criticidad de los datos que se pretenden manejar, se ha decidido utilizar claves AES 128 bits puesto que Wifi, Bluestooth o ZigBee soportan dicha encriptación. Es un sistema de encriptación que no es factible de ser roto por medio de fuerza bruta y adecuado para sistemas de desafío respuesta para la función de autenticación. Los dispositivos de la red, que son los más limitados computacionalmente soportan dicha encriptación lo que convierte a esta encriptación adecuada para nuestro entorno.

Como se ha dicho en la etapa de investigación, el uso de clave simétricas dotan de un alto de seguridad al sistema, aunque los métodos de negociación para la generación de las claves de sesión con incompatibles con los dispositivos IoT. Por esta razón, se opta por un modelo de distribución de dichas claves en base a un componente de la red que maneja dicha información, sea PM o SA. Durante el proceso de instalación se definirán dichas claves, y una vez superado este estado, las claves solo podrán circular por la red encriptados sobre claves simétricas de otros dispositivos como demostraremos en el apartado de mensajería 4.3.

4.3.2 Autenticación

Se trata de del primer paso para crear una relación de confianza entre los dispositivos del sistema para securizar el intercambio de información. Una vez más dependiendo del dispositivo la manera de almacenar dicha informaron puede ser muy dispar entre dispositivos. Los puntos finales de las redes de IoT / M2M debe ser autenticados sin intervención Humana por tanto sistemas de user-password tradicionales no puede ser utilizados en estos sistemas de manera eficiente y puede requerir re-autenticación. Por tanto se utilizan sistema que incluyen la identificación por radiofrecuencia (RFID), secreto compartido, certificados X.509 (ya descartado), la dirección MAC del punto final, o algún tipo de hardware basado un nodo raíz totalmente confiable, nodo raíz inmutable.

El modelo de Autenticación de dispositivos básicamente es el modelo de desafío - respuesta con la salvedad de que no circula el desafío en plano por la red. El dispositivo debe cifrar una serie de datos para verificar su identidad. En caso de los dispositivos deberá cifrar correctamente los códigos de operación, su IP, un identificador único definido por la propia pasarela a la que se conecta y el número de secuencia. En caso de la comunicación entre pasarelas, estas deberán cifrar correctamente los códigos de operación/respuesta además de su IP o MAC. Ningún elemento malicioso de red que pretenda conectarse podrá intervenir en

nuestra red pues no conocerá ninguna de las claves que intervienen en la transmisión de información. Semejante al modelo del sistema Kerberos en la que se transmite información entre dispositivos sin que los mismos puedan descifrar datos críticos.

La Pasarela Maestra y el Servidor de Autenticación tienen la asignación de una clave en función de la dirección IP definida para los dispositivos. Por tanto se debe tener especial atención a la asignación de las IPs a los dispositivos.

Por otro lado, la instalación de dispositivos debe hacerse de manera manual, por tanto para poder acceder a ellos, todos los componentes de la red y los datos que albergan están protegidos por usuario y contraseña.

El proceso de instalación y migración en los que está contemplada la función de autenticación se explicada en los puntos 4.3.1 y 4.3.3

4.3.3 Integridad – Firma

En caso de que los dispositivos quieran firmar los datos que envían, para verificar su identidad e integridad, además de su clave de cifrado, deberán almacenar su clave privada de firma. En caso de que sea necesaria la firma de los datos, esta funcionalidad aumentará la necesidad de recursos de los dispositivos *edge* de la red.

4.4 Estados y mensajería

En el desarrollo del proyecto se han contemplado una serie de estados por los que va a pasar el sistema y la estructura de la mensajería entre dispositivos ligado a dichos estados.

1) Instalación
2) Envió de datos
3) Migración de dispositivos entre pasarelas
4) Desconexión
5) Reconexión en caso de pérdida de señal
6) Tareas de Actualización

Con el fin de evitar la sobrecarga de información que provoca TCP, se ha decidido utilizar UDP como protocolo de transmisión. De esta manera se disminuye la longitud de las tramas, y se selecciona cuales de ellas necesitan ACK para su funcionamiento. Por el medio del uso de UDP se intenta limitar lo máximo posible la complejidad de la mensajería para adaptarse a las necesidad de los dispositivos del *edge* de la red.

Se han definido una serie de códigos para identificar las distintas peticiones y respuesta de nuestro entorno.

Código	Significado
100	Confirmación de Instalación
101	OK Instalación
200	Envío de datos
201	ACK datos
300	Solicitud de conexión
301	ACK conexión
400	Solicitud de migración

401	Migración OK
402	Migración NOK
500	Aviso de Desconexión
600	Solicitud de Actualización de Estado
601	Actualización OK
602	Actualización NOK

Tabla 1: Códigos para la identificación de mensajería

4.4.1 Instalación

Durante el proceso de instalación se despliega el sistema y los distintos elementos que van a formar nuestra red. Como se ha explicado en la definición de la arquitectura, la instalación de los dispositivos se realiza en la Pasarela Maestra o en el SA. Se define un identificador único de red para el dispositivo y se le asigna un número de secuencia aleatorio. Dichos datos son introducidos en el dispositivo a la hora de arrancarlo para que su solicitud de entrada en la red pueda ser confirmada por el SA/PM.

La instalación de los dispositivos como su aceptación en la red es una de los pasos que difieren dependiendo del entorno en el que nos situaremos, por tanto procederemos a explicar ambos entornos.

EL modelo desafío - respuesta reside en que el dispositivo debe codificar un serie de información correctamente que solo conoce la PM o SA: Dirección origen asignada al dispositivo, Identificador único y numero de secuencia.

4.4.1.1 Entorno Domestico

En el entorno más pequeño, el entorno domestico, se accede a la PM y se introduce el dispositivo que se va a arrancar. A continuación se arranca el Dispositivo y se le introduce la PM de nuestra red para que mande la confirmación de Instalación y entre a formar parte de nuestra red.

Figura 9. Diagrama del proceso de Instalación en entorno domestico

Se construye una trama UDP con la siguiente configuración en la que los campos soberados en azul están cifrados con la clave del dispositivo y son los campos chequeados en la PM para verificar la identidad del dispositivo. En este caso toda la información va cifrada salvo

el código que identifica la trama como trama de dispositivo, no es necesario cifrar esta información pues no es un dato sensible.

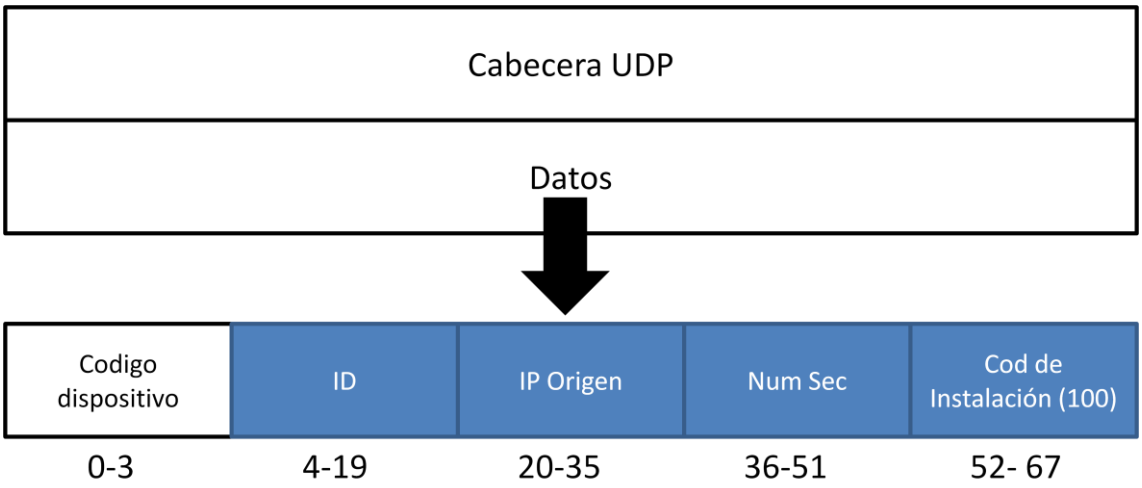

Figura 10. Traza de Solicitud de Instalación. Sombreado azul cifrado.

Los bloques se cifran de forma independiente y de igual manera son descifrados de manera independiente. En caso de descifrar alguno de los bloques de forma incorrecta, o la información descifrada con coincide con los datos almacenado en la PM, el dispositivo no puede entrar a formar parte de la red.

Las PE instaladas en el entorno ya contienen los datos de la PM necesarios para operar, por tanto no es necearía una confirmación para su despliegue. Todos los dispositivos deben ser desplegados en el PM y desde ese punto migraran hacia la PE que se precise. Se explicara el proceso de migración con más detalle en el aparatado 4.3.3. Además las tablas de pasarelas del entorno se actualizaran mediante las tareas de mantenimiento explicadas en el apartado 4.3.5

4.4.1.2 Entorno Local.

En un entorno de mayor tamaño se opta por la opción de la utilización de un SA. Las PE se desplieguen dotadas de la clave necesaria para comunicarse con el SA, introducida por el administrador de la red. A la hora de desplegar un dispositivo, la PE donde se instala el dispositivo funciona como un servidor proxy redireccionando la petición de confirmación de instalación. Dicha pasarela solo el capaz de leer el código de dispositivo y el código de instalación pues desconoce la clave para descifrar los datos sensibles.

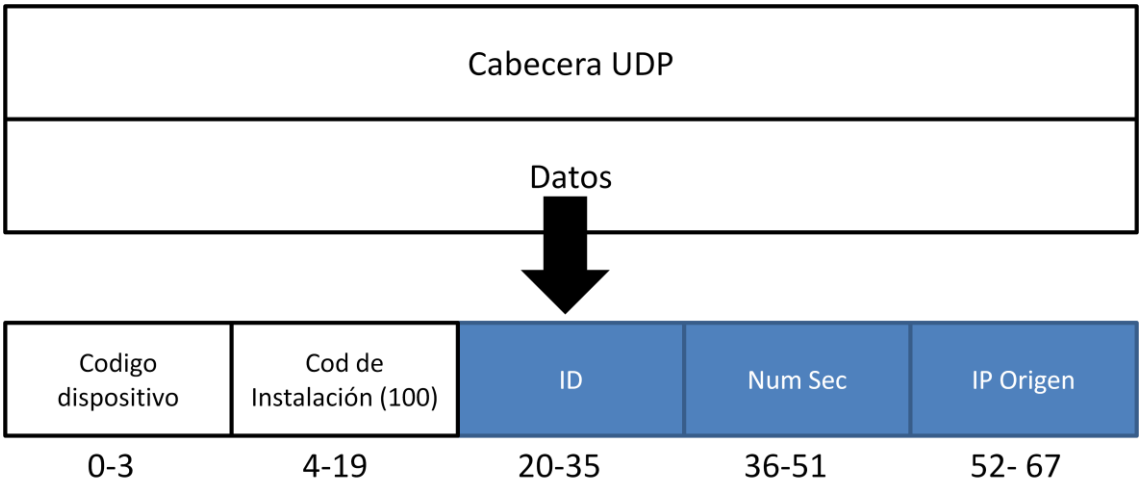

Figura 11. Trama de solicitud de instalación de un dispositivo en entorno local. Los datos sombreados en azul (codificados) son ilegibles para la pasarela pues no conoce la clave del dispositivo

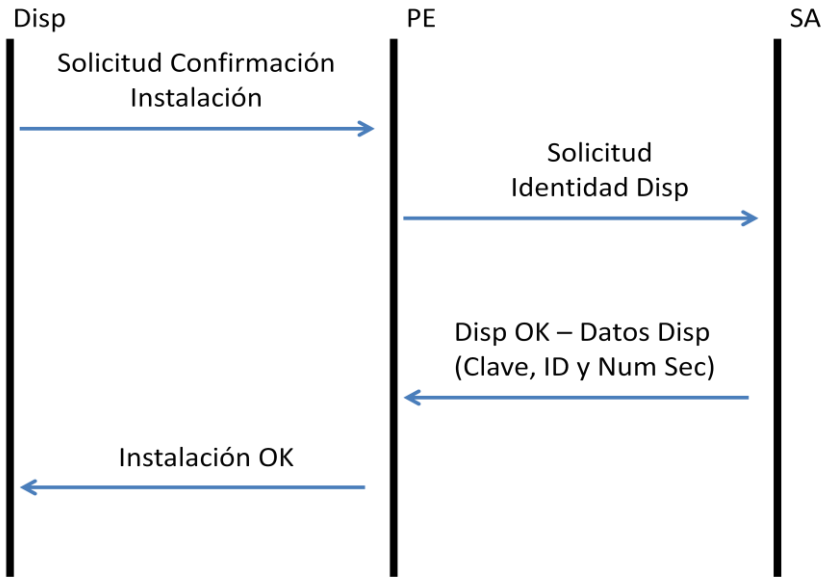

Figura 12. Diagrama del proceso de Instalación en entorno local

Cuando el SA confirma la identidad del dispositivo, cifrando con la clave de la PE, le envía la clave, el identificador único y el número de secuencia Dispositivo para que la PE y el Dispositivo se puedan comunicar. De igual manera, ahora que ya conoce la clave, el PE utiliza la clave del Dispositivo para enviarle su clave y de esta manera de pueda comenzar la transmisión de datos.

En caso de descifrar alguno de los bloques de forma incorrecta, o la información descifrada con coincide con los datos almacenado en la SA, el dispositivo no puede entrar a formar parte de la red

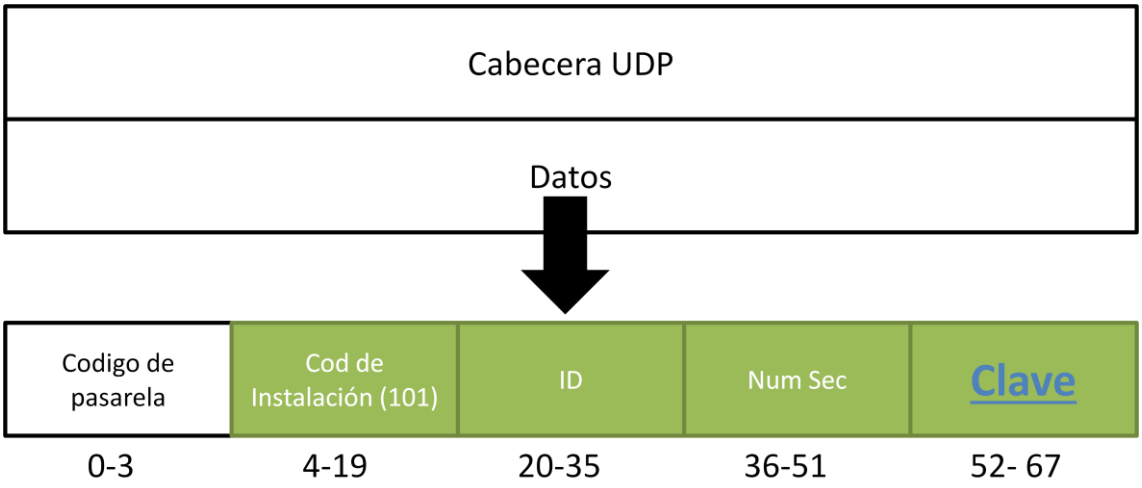

Figura 13. Traza de confirmación en el proceso de instalación (SA->PE). Traza cifrada con la clave entre pasarela y SA en que se encuentra la clave del dispositivo

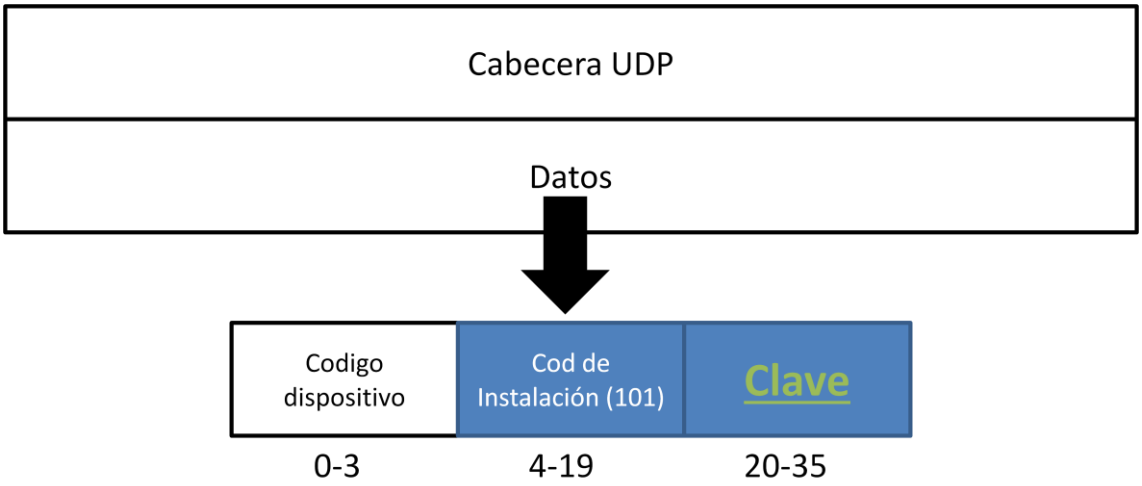

Figura 14. Traza de confirmación en el proceso de instalación (PE->Dispositivo). Traza cifrada con la clave del dispositivo en la que se encuentra la clave de la pasarela.

4.4.2 Envío de datos

El envío de los datos es idéntico en todos los entornos pues es un proceso entre la pasarela y el dispositivo una vez haya confirmando la instalación de todos los dispositivos. Una vez distribuidas las claves la transmisión de datos puede producirse. Dependiendo de la criticidad de los datos que se quieren mandar, en este punto se puede optar por la utilización de ACK para la confirmación de la recepción de los datos. Las tramas se construirían de la siguiente manera.

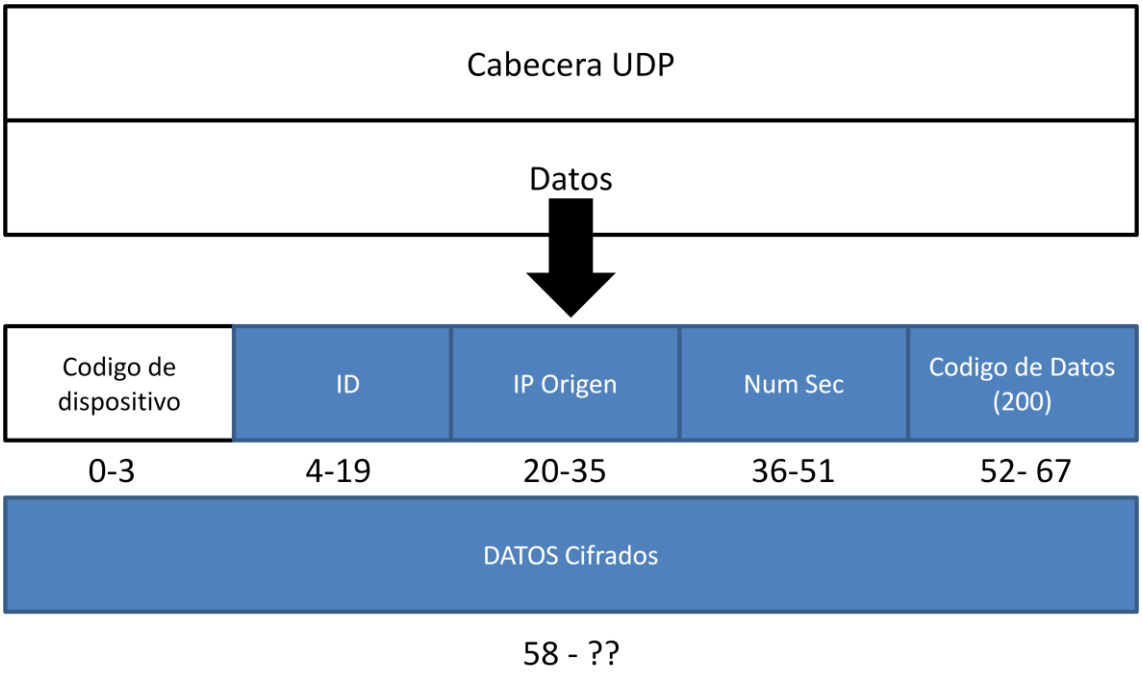

Figura 15. Traza de envío de datos.

En caso de ser necesario el uso de un ACK se codifica una trama con el código 201 desde la PE-PM al dispositivo.

Se ha diseñado el sistema para que, en caso de ser necesario un ACK para la confirmación del envío si se está manejando datos de alta sensibilidad, se reenvié la trama 3 veces en caso de no recibir el ACK pasado un cierto tiempo. En ese caso de ser negativa la recepción del ACK después de los reintentos, el sistema activará una alerta para informar al usuario y este procederá a solucionar manualmente el problema por medio de un reconexión o reinstalación del dispositivo.

4.4.3 Migración

La migración entre pasarelas es otro punto en el que el procedimiento difiere dependiendo del entorno en el que nos encontremos. El periodo de migración de un dispositivo móvil se produce cuando comienza a recibir señal de otra de las pasarelas de la red y la conexión a la misma mejora la fiabilidad en la recepción de la señal, se sigue el criterio de *Best-Connected*, es decir, conectado al dispositivo que garantice la mejor recepción de señal.

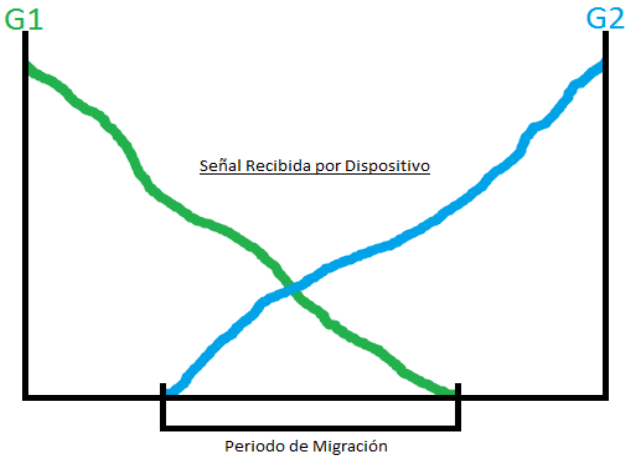

Figura 16. Diagrama del periodo de migración de dispositivos entre pasarelas.

4.4.3.1 Migración en entorno domestico

Cuando un dispositivo se aleja de la pasarela a la que está conectado y recibe una señal de una pasarela vecina comienza el proceso de migración. Las pasarelas de un mismo entorno se conocen entre sí, es decir, que conocen sus claves, direcciones y puertos de escucha. Durante el proceso de instalación y las tareas mantenimiento se han mantenido actualizada estas tablas.

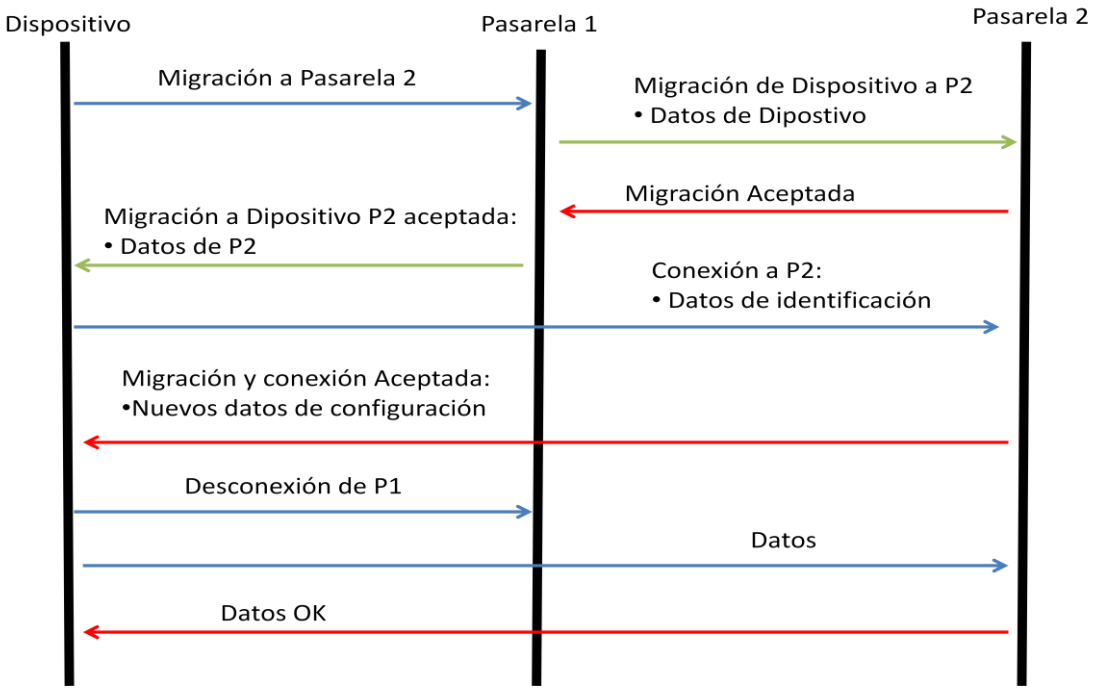

Figura 17. Diagrama de migración en entorno domestico.

El dispositivo solicita la migración a la pasarela 2 (G2) a través de la pasarela 1 (G1) y se queda a la espera de una contestación. Se activa un temporizador en para marcar el final de esta etapa de espera.

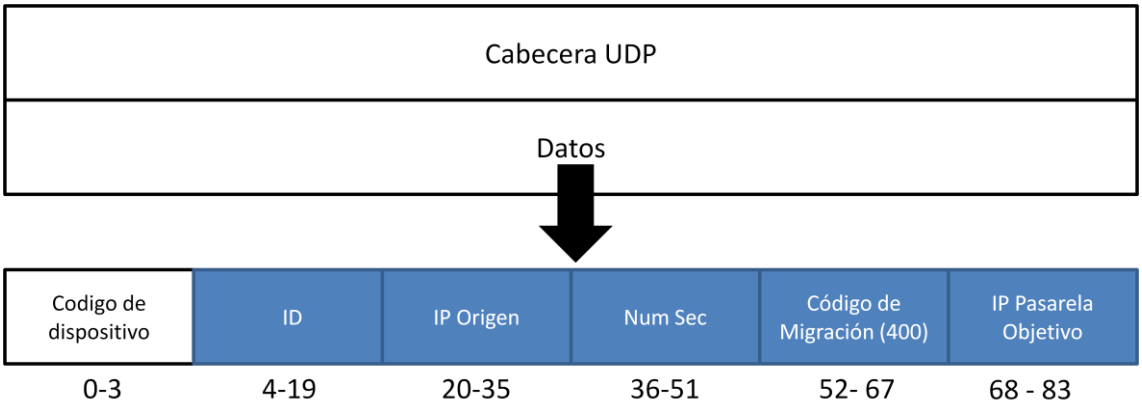

Figura 18. Traza de Solicitud de migración (Dispositivo -> Pasarela)

Tras la verificación de los datos correspondientes explicadas en apartados anteriores, en caso de ser G2 una pasarela valida (conocida por G1), G1 le redirecciona la petición a G2 con los datos de identidad del Dispositivo.

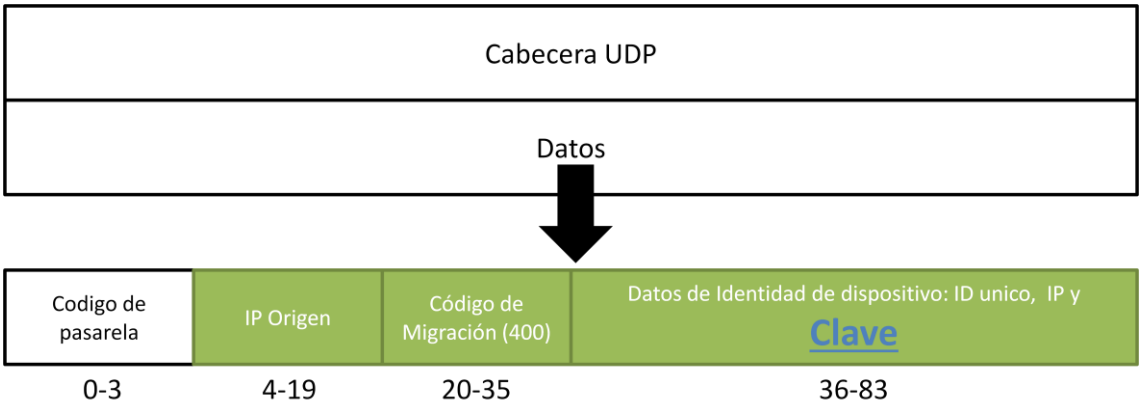

Figura 19. Traza de Solicitud de migración (Pasarela origen -> Pasarela destino)

Una vez la pasarela G2 ha añadido el dispositivo, devuelve la confirmación (código 401) a la G1 (cifrado con su clave). G1 a su vez redirecciona la confirmación junto a los datos de la pasarela G2 al dispositivo, datos que deberían llegar antes de que se produzca el *timeout* del temporizador activado al principio del proceso.

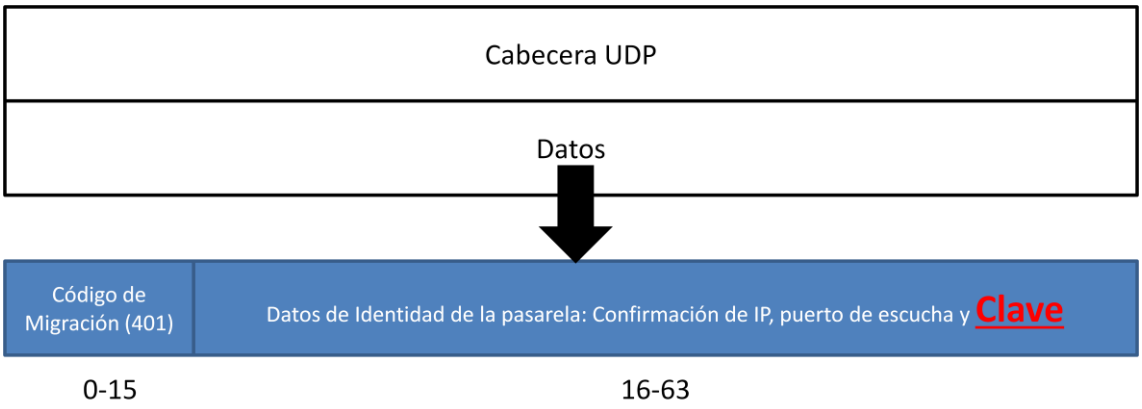

Figura 20. Traza de confirmación de migración (Pasarela origen -> Dispositivo). La traza contiene la clave (cifrada sobre la clave que comparten la pasarela origen y el dispositivo) necesaria para que el dispositivo se comunique con la nueva pasarela objetivo a la que desea migrar

En este punto se produce la solicitud de conexión del dispositivo a dicha pasarela. El dispositivo conociendo la clave de G2 le envía una solicitud de conexión (código 300) con sus datos de identidad: Ip origen cifrada, numero de secuencia (debe ser 0 para esta solicitud) e identificador único. G2 conociendo la clave del Dispositivo descodifica la información y genera un nuevo identificador único dentro del rango de criticidad anterior y un nuevo número de secuencia y se los envía de vuelta al dispositivo para que reconfigure sus paramentaros de identidad para que la transmisión de datos pueda producirse bajo los nuevos parámetros.

En este punto se vuelve a activar un temporizador pues el dispositivo se vuelve a quedar a la espera de la respuesta de la nueva pasarela. En caso de que se active el *timeout* se reintenta la conexión un número de veces. En caso de no poder producirse la conexión se activara una alarma en el dispositivo pues podría quedar sin conexión a la red.

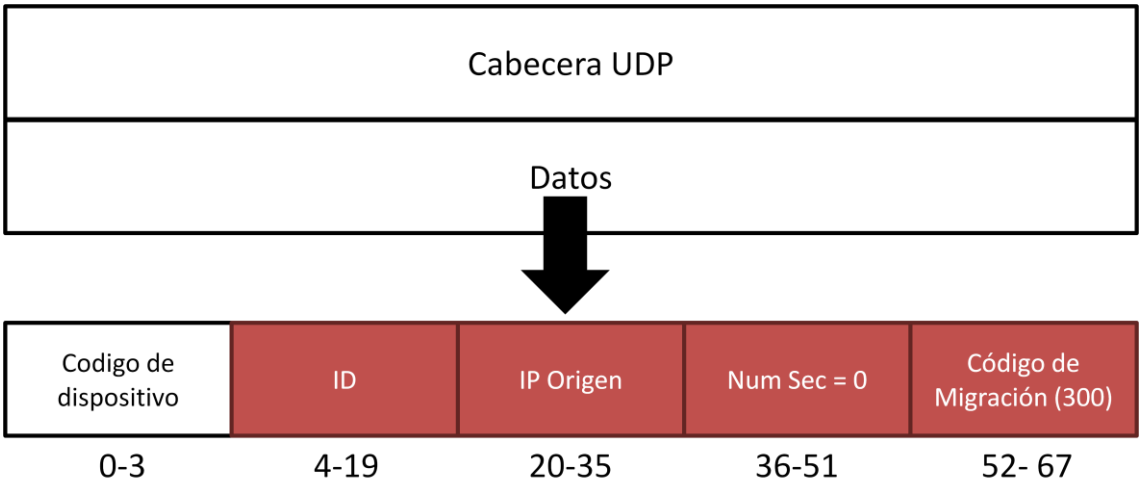

Figura 21. Traza de Solicitud de Conexión

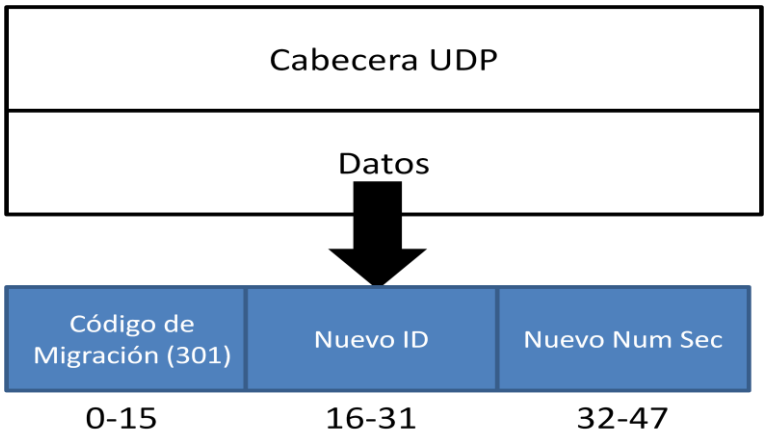

Figura 22. Traza de Aceptación de Conexión

Una vez conectado a G2, el dispositivo envía un orden de desconexión a G1 (código 500), reconfigura sus parámetros de identidad y comienza la transmisión de datos hacia G2. La orden de desconexión no requiere ACK, si no se recibe en el dispositivo, gracias a las marcas de tiempo registradas en G1, durante las tareas de mantenimiento de desactivara el dispositivo.

4.4.3.2 Migración en entorno local

En caso de que el dispositivo solicite una migración entre desde G1 a G2, si G1 y G2 se conocen el proceder es idéntico al caso domestico. En caso de que no se conozcan es necesario realizar una consulta al SA para verificar la identidad de G2. Si se confirma la identidad de G2, ambos G1 y G2 guardaran los datos uno del otro para posteriores migración. Las tareas de mantenimiento se encargaran de mantener actualizada la información que contienen las pasarelas sobre las pasarelas de su entorno.

La manera de proceder en caso de que G1 y G2 no se conozcan y por tanto G1 no tenga la información necesaria para conectarse a G2 seria:

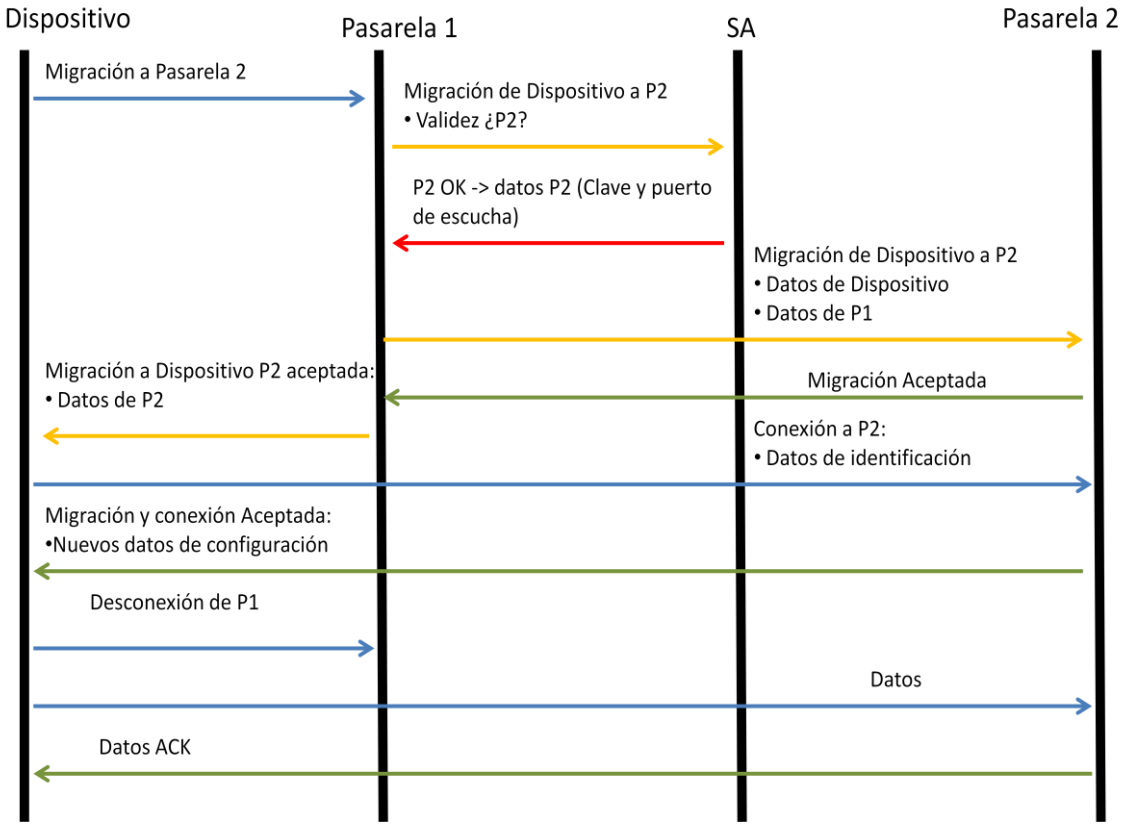

Figura 23. Diagrama de migración en entorno local- p.31

4.4.4 Desconexión

Una vez conectado a G2, el dispositivo envía un orden de desconexión a G1 (código 500), reconfigura sus parámetros de identidad y comienza la transmisión de datos hacia G2. La orden de desconexión no requiere ACK, si no se recibe en el dispositivo, gracias a las marcas de tiempo registradas en G1, durante las tareas de mantenimiento de desactivara el dispositivo.

La desconexión puede ser enviada en cualquier momento, no solo tras una migración, pero como en todos los chequeos, solo se hará efectiva si los datos e identidad son correctos.

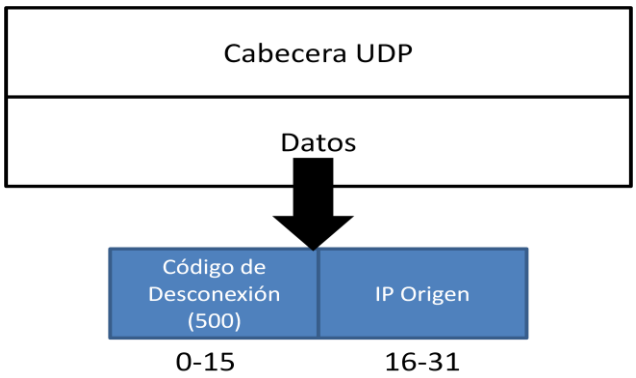

Figura 24. Traza de Informe de Desconexión

4.4.5 Reconexión

El proceso de reconexión se produce cuando un dispositivo ha perdido cobertura, o ha abandonado el entorno donde opera y necesita volver a conectarse. Idéntico que el proceso de conexión tras un migración, a través de la PM en el entorno doméstico o través de una PE (función proxy) haciendo una solicitud al SA se envía una orden con código 300 y se reconfigura los parámetros de entidad del dispositivo y la transmisión de datos se retoma.

4.4.6 Tareas de Mantenimiento

Las tareas de mantenimiento son las encargadas de mantener la “buena” salud de la red. Por medio de la programación de dichas tareas se mantiene el correcto funcionamiento e independencia la red.

4.4.6.1 Actualización de dispositivos en la pasarela.

Esta tarea estará encargada actualizar los elementos que mantiene en memoria una pasarela. Una pasarela tiene una serie de Dispositivos y pasarelas vecinas en memoria para minimizar las consultas al SA.

1) Dispositivos

En caso de los dispositivos, la pasarela registra la fecha del último intercambio de información. Definiendo una ventana de tiempo se seguridad, todos los dispositivos inactivos a partir de esa ventana temporal serán eliminados de la pasarela. De esta manera, la gestión de dispositivos y sus claves entre pasarelas en procesos de migración podrá ser más eficiente pues la búsqueda entre elementos disminuye. Además, en caso de que una pasarela quede comprometida, de esta manera quedaran afectados en menor número de dispositivos.

2) Pasarelas

En el caso de las pasarelas, se utilizara el envío de una solicitud de actualización (código 600) para verificar que las pasarelas en memoria siguen activas. Si este código es contestado con un código 601, se mantendrá la pasarela en memoria, mientras que si se agota el temporizador de espera de respuesta y una serie de reintentos, se notificara a la SA o PM con código 602 y la dirección de la pasarela con la que se ha obtenido el conflicto.

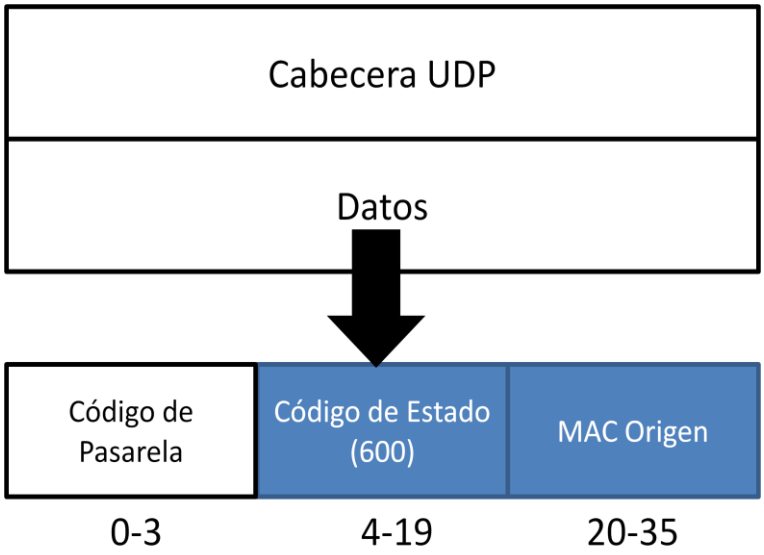

Figura 25. Traza de Solicitud de informe de estado.

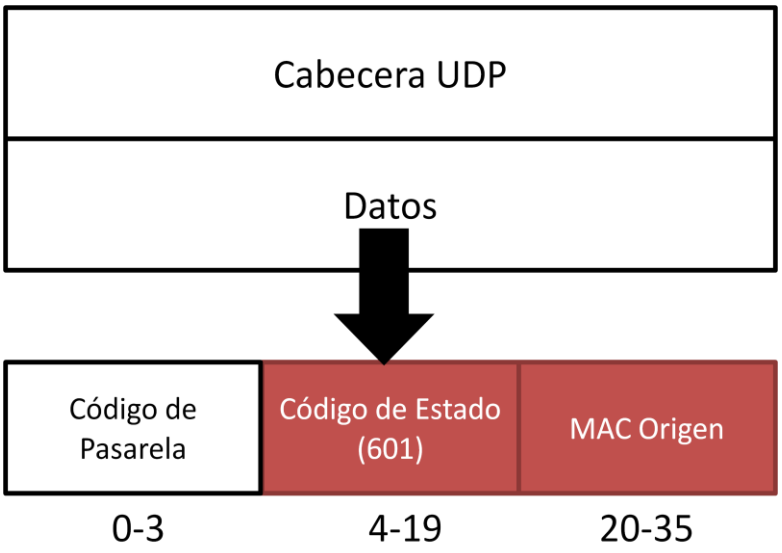

Figura 26. Traza de Confirmación de informe de estado.

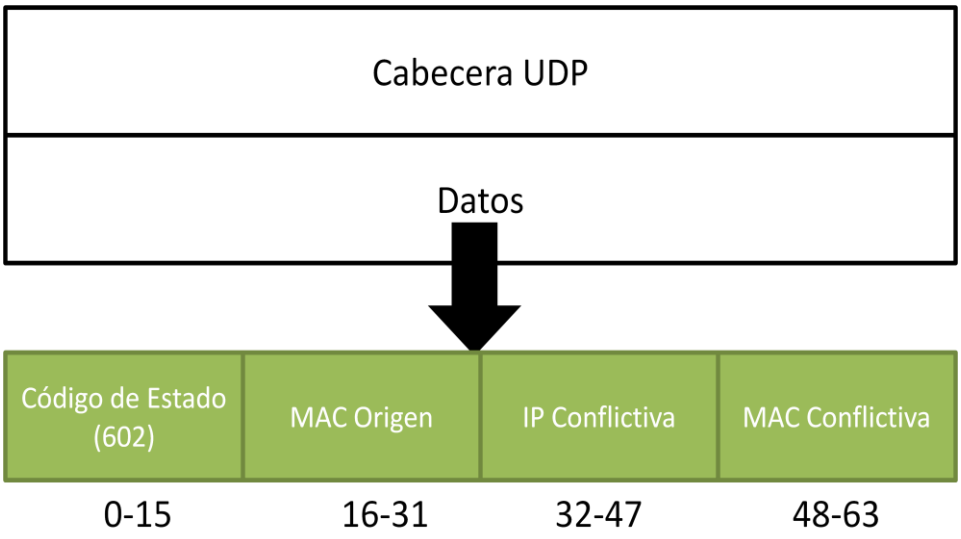

Figura 27. Traza de Informe de Pasarela Conflictiva

Esta tarea de mantenimiento puede programarse manera automática y cíclica. Cada X tiempo puede ejecutarse y proporcionar un informe sobre la salud de la red al administrador o controlador de la misma. Si se observar un gran número de pasarelas marcadas como conflictivas, esto requeriría de un análisis más profundo.

Capítulo 5. Desarrollo de Software – Implementación

Una vez completada la fase de diseño de nuestro software, se va a llevar a cabo el desarrollo de un caso de uso. En vista de los recursos a lo que se tiene acceso, se va a llevar a cabo el desarrollo del entorno domestico, formado por la Pasarela Maestra, Pasarela Estándar y Dispositivos IoT. Para el desarrollo de los componentes que van a formar parte de nuestro entorno se va a utilizar JAVA, pues es un lenguaje de programación adecuada para el uso de comunicaciones sobre redes de internet debido a las clases y métodos que ya implementa dicho lenguaje de programación.

Una vez realizada la implementación de los dispositivos, se llevara a cabo una prueba, en la que por medio de Wireshark, se captura el tráfico asociado a nuestro sistema y se analizara el contenido de dicha mensajería para contrastar la seguridad del entorno asociado.

A continuación se repasaran las etapas definidas en el apartado de diseño y el flujo de mensajería capturado en cada etapa.

Los mensajes construidos durante la implementación, además de los datos cifrados, contienen la información sin cifrar para mostrar el contenido que transporta cada mensaje. Los datos de texto capturados al final de cada mensaje en el modelo final NO existe.

5.1 Instalación.

Se han definido 3 elementos y sus atributos durante el proceso de instalación: 1-PM, 1-PE y un dispositivo.

PM – IP: 192.168.1.6

PE – IP: 192.168.1.4

Dispositivo – IP: 192.168.1.2

Se han definido los códigos de dispositivo (código 12345) y pasarela (código 54321) para conocer la procedencia del mensaje. Los atributos de identidad asociados a los componentes anteriores se definen a continuación, pues dicho atributos iniciales solo son conocidos por el administrador y son punto clave para securizar la red. Además la asignación de dichas IP también ha sido realiza por el administrador, por tanto cualquier elemento con IP desconocida o fuera de las definidas por el administrador quedará su trafico descartado.

5.1.1 Pasarela Maestra

La PM definida contiene la información de los componentes de nuestra red domestica, accediendo a dicho dispositivo mediante usuario y contraseña y se han definido os siguientes componentes. Un vez fijado los atributos de identidad para el dispositivo como son el ID y el

Num Sec, estos son también introducidos en la fase de instalación de dicho dispositivo para que pueda identificarse.

Pasarela Estándar:	Dispositivo
IP: 192.168.1.4	ID: 1111
Puerto de Escucha: 7777	IP: 192.168.1.2
Clave: Gateway222222222	Clave: IotDevice1111111
	Num Sec: 23

Tabla 2: Datos de Instalación de la Pasarela Maestra

La PM se ha definido sin limitaciones computaciones pues es un equipo que gozara suficientes atributos de hardware para realizar todas sus tareas.

5.1.2 Pasarela Estándar

La PE definida tiene todos los atributos de la PM salvo el repositorio de dispositivos, por tanto solo hay que definir en dicho dispositivo, tras identificarse adecuadamente con usuario y contraseña de administrador, los atributos de la PM:

Pasarela Maestra
IP: 192.168.1.6
Puerto de Escucha: 5555
Clave: Gateway111111111

Tabla 3: Datos de Instalación de la Pasarela Estándar

5.1.3 Dispositivo

Para emular un dispositivo IoT se ha definido un Maquina Virtual en un equipo en la que los atributos de dicha maquina virtual intenta emular las limitación de hardware y por tanto de software de elementos IoT. Se le ha atribuido a dicha maquina 1 GB de RAM, 5 GB de almacenamiento para realizar las tareas necesarias. Una vez realizado la introducción de la PM en el Dispositivo, se confirma el proceso de instalación enviando el código 100 y siendo respondido por el código 101. Para el intercambio de información, el dispositivo ha abierto el puerto número 56128.

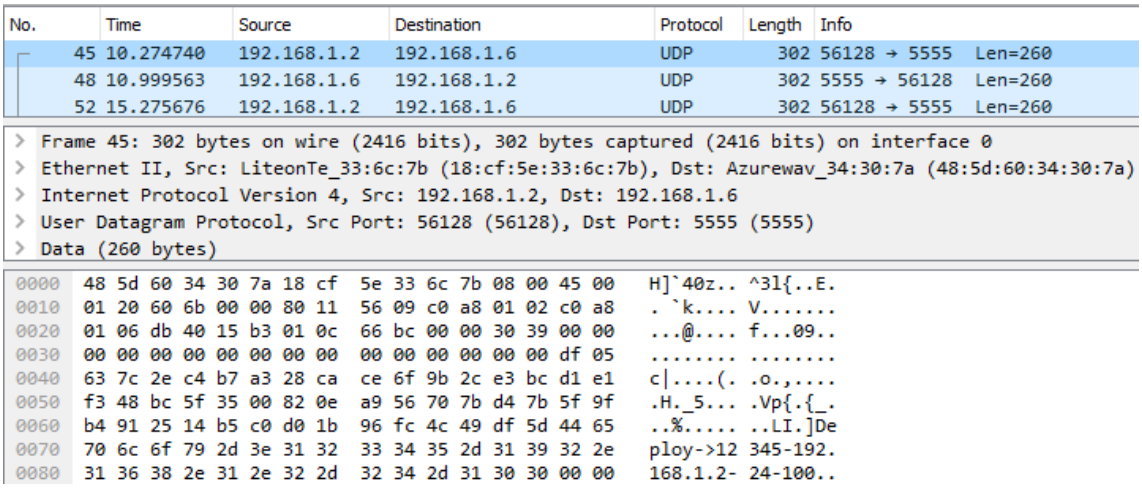

Figura 28. Captura de Wireshark: Confirmación de instalación con código 100 desde el dispositivo a la PM junto a los atributos de identidad definidos durante la instalación.

En la figura 28 observamos la trama de solicitud de conexión con la red del Dispositivo (192.168.1.2) hacia la PM (192.168.1.6). El dispositivo ha abierto un puerto cualquiera (56128) para la conexión, no hay necesidad de definir dicho puerto, puede ser cualquiera. Por otro lado la PM ha, definido por el administrador escucha peticiones en el puerto 5555, es ahí donde se envía esta solicitud. Como se observa en la figura, la información para su identificación (ID, ip origen y numero de secuencia) y el código de instalación 100 van cifrados con la clave dispositivo-PM.

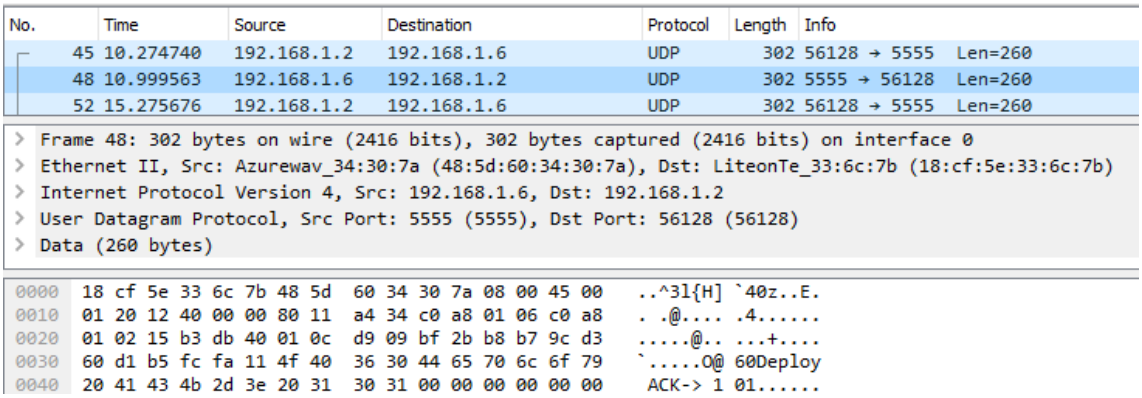

Figura 29. Captura de Wireshark: Confirmación de instalación con código 101 desde la PM al dispositivo.

En la figura 29 se observa la confirmación de la instalación para que le dispositivo pueda empezar a transmitir. La información va cifrada con la clave dispositivo-PM, con origen en la PM (162.168.1.6) con puerto 5555 y destino el Dispositivo (192.168.1.2) con puerto 56128.

5.2 Envío de Datos.

Una vez confirmada la instalación del dispositivo, se puede proceder al envío de datos entre Dispositivo y Pasarela. Se ha activado el uso de ACK para el envío de datos por parte del dispositivo. Se ha introducido un mensaje genérico al paquete además del número de secuencia junto a ese mensaje para emular los datos enviados por el dispositivo:

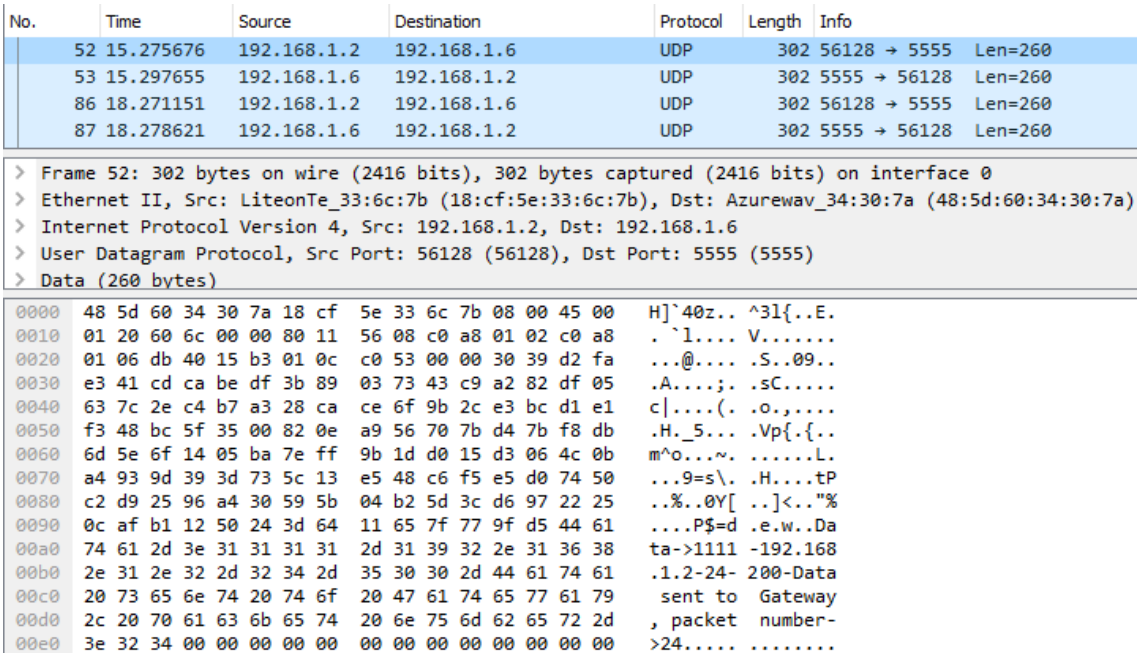

Figura 30. Captura de Wireshark: Primer envío de datos (código 200) con atributos de identidad y el mensaje “Data sent to Gateway, packet number->24”

En la figura 30 se observa la traza utilizada para el envío de datos entre el dispositivo IoT y la pasarela, en este caso PM. Con origen el dispositivo (192.168.1.2) y puerto 56128 y destino la PM (192.168.1.6) con puerto 5555 se envía cifrado con la clave del Dispositivo los parámetros de identidad y autenticación (ID, Ip Origen y Sec Num) y un mensaje genérico que emula los datos a trasmitir: “Data sent to Gateway, packet number->24” el

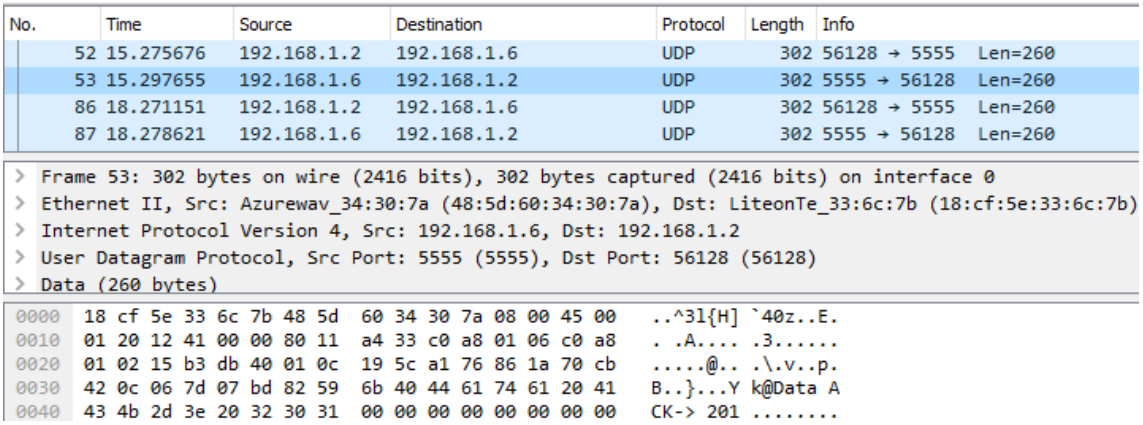

Figura 31. Captura de Wireshark: ACK de datos (código 201) al primer envío de datos.

En la figura 30 se observa la traza de confirmación del envío de datos con código 201 pues se ha marca este tráfico como crítico y el ACK es necesario para asegurar la recepción de datos. EL código viaja de la PM (192.168.1.6) del puerto 5555 al Dispositivo (192.168.1.2) con puerto destino 56128 cifrado con la clave de la PM.

A continuación se muestra otro ejemplo de envío de datos y su ACK correspondiente en el que se ve que el número de secuencia ha aumentado en 1, cumpliendo los parámetros de seguridad.

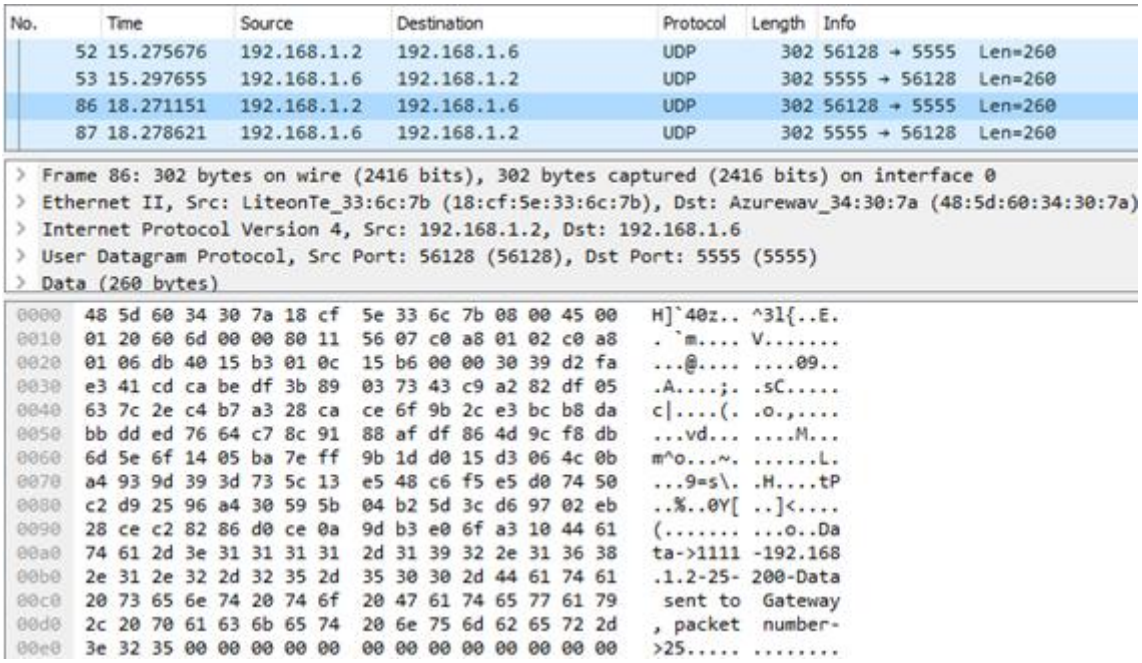

Figura 32. Captura de Wireshark: Segundo envío de datos (código 200) con atributos de identidad y el mensaje “Data sent to Gateway, packet number->25”

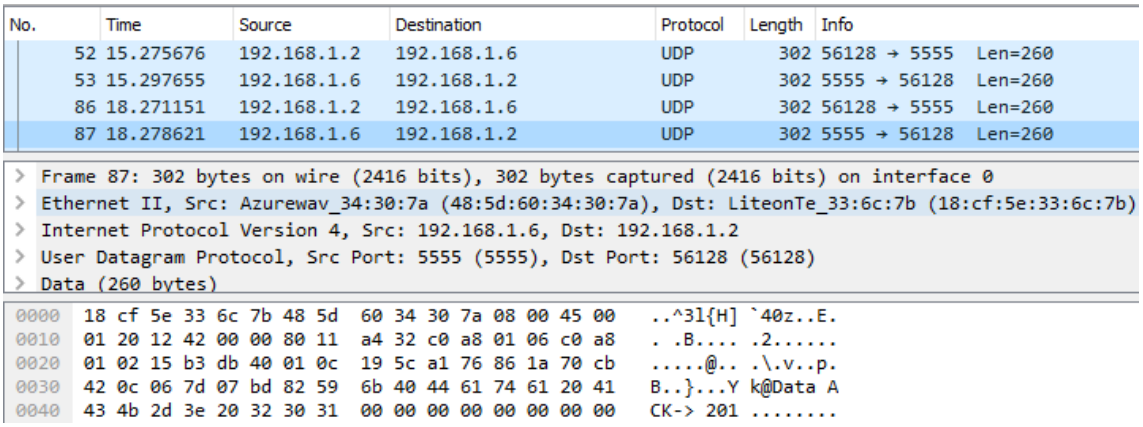

Figura 33. Captura de Wireshark: ACK de datos (código 201) al segundo envío de datos.

5.3 Migración

La migración es uno de los puntos críticos del sistema que se ha desarrollado, en este proceso intervienen todo los componentes y durante este proceso se produce el intercambio de datos necesario (claves, IDs, etc) para hacer posible la comunicación entre los mismos. Esta compuesta por varios pasos que se explicaran por medio de la captura de WireShark:

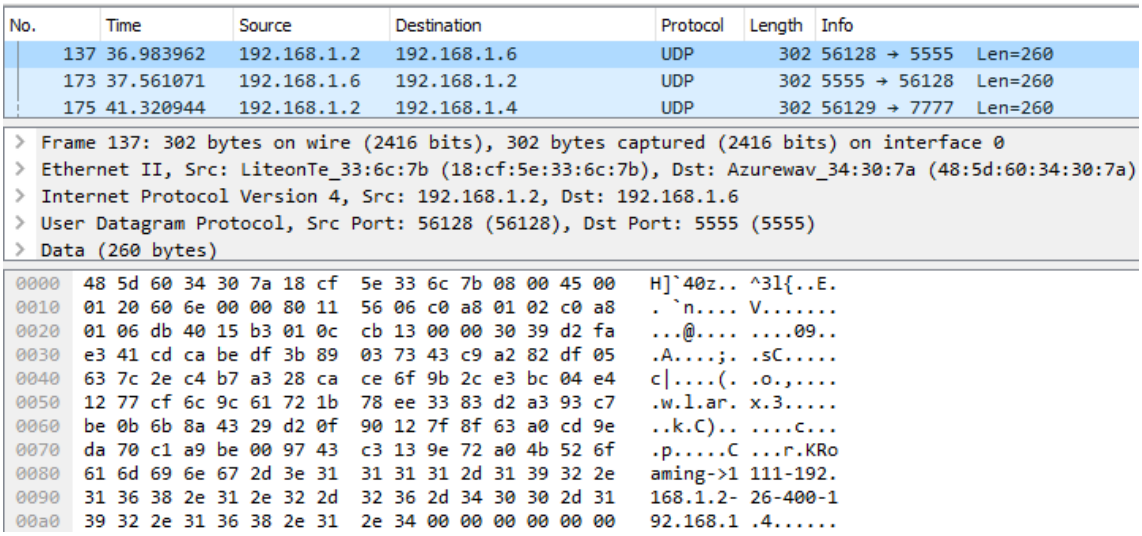

Figura 34. Captura de Wireshark: Solicitud de Migración (código 400) a PE (192.168.1.4) desde el Dispositivo.

En la figura 34 se observa la traza de solicitud de migración desde el Dispositivo (192.168.1.2) con puerto 56128 a la PM (192.168.1.6) con puerto 5555. Al captar señal desde la pasarela 192.168.1.4 y disminuir la señal procedente de la PM (192.168.1.6) comienza el proceso de migración por medio de este mensaje con código 400. Como el resto de mensajes contiene los datos de identidad (ID, Ip y Num Sec) además, en el campo de datos, la Ip de la pasarela a la cual quiere migrar. Este mensaje una vez más va cifrado con la clave del Dispositivo.

Una vez PM ha comprobado que PE es una pasarela valida, le envía la información necesaria para que PE identifique al dispositivo y espera la respuesta. Para el proceso de migración de dispositivos, PM abre un nuevo puerto para trasferir esta información a PE y donde espera la respuesta.

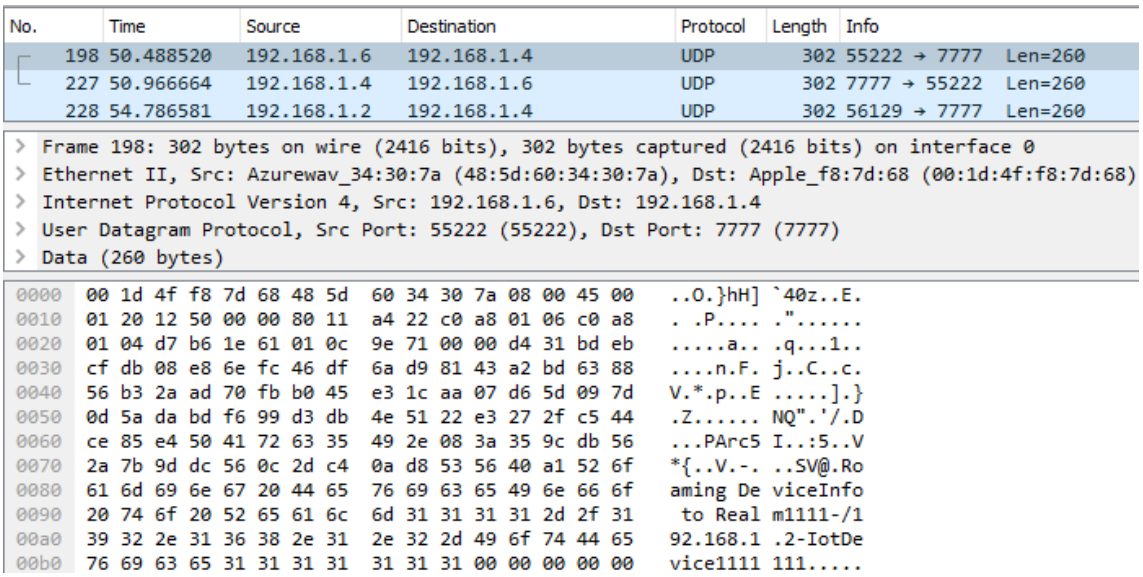

Figura 35. Captura de Wireshark: Solicitud de Migración (código 400) con los datos del dispositivo que desea migrar de PM a PE.

En la figura 35 se observa la traza con enviada desde PM (192.168.1.6) a PE (192.168.1.4). Hay que destacar, como se ha explicado anteriormente, que PM envía los datos al puerto 7777 de PE pero desde un puerto abierto para este tipo de gestión, este puerto es aleatorio y en este caso es el 55222. Es en este puerto donde PM esperara la respuesta a esta petición de migración. En el mensaje se observa los datos necesarios para la identificación del dispositivo que desea migrar: ID, Ip y clave necesaria para desencriptar la información procedente del mismo. Dicha información esta cifrada con la clave entre pasarelas.

Una vez PE ha añadido el dispositivo a su sistema, manda la confirmación a PM y este a su vez reenvía esa información al Dispositivo junto a los datos necesarios para comunicarse y autenticarse en PE.

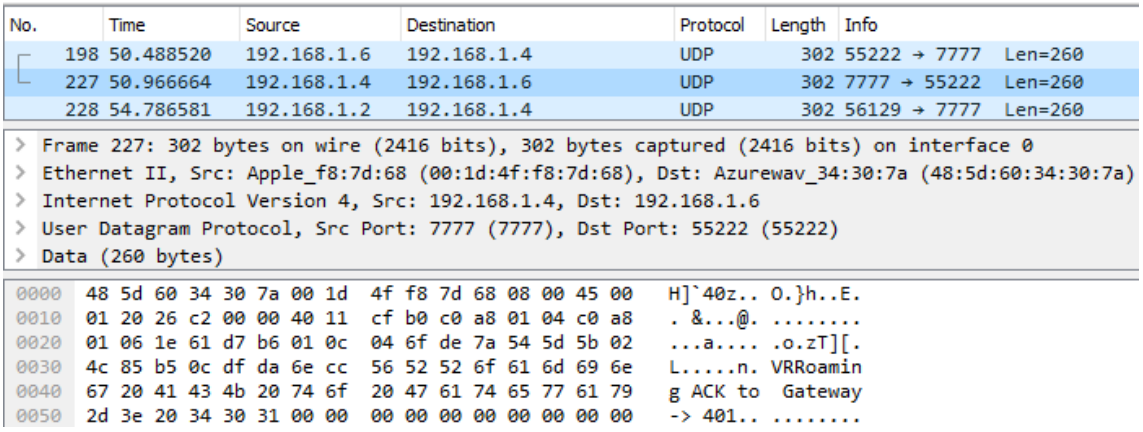

Figura 36. Captura de Wireshark: ACK de Migración (código 401) de PE a PM.

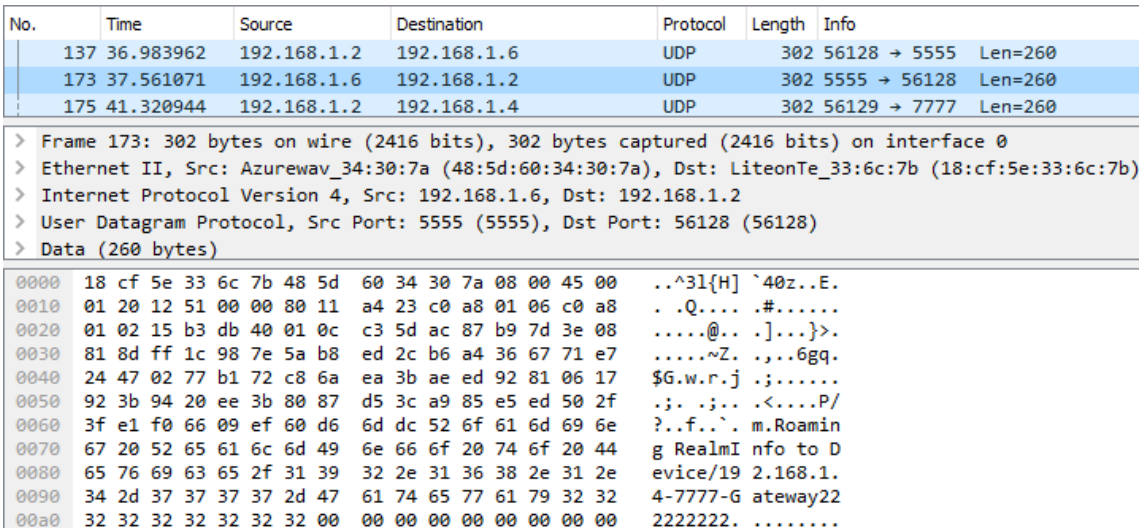

Figura 37. Captura de Wireshark: Confirmación de Migración (código 401) con los datos necesarios para que el dispositivo se comunique con la pasarela de PM al Dispositivo.

En las figuras 36 y 37 se observa el proceso de confirmación de migración. En la figura 37 se envía la confirmación desde PE a PM. Como se ha dicho, dicha respuesta debe ir dirigida al puerto que ha abierto PM para gestionar la migración del dispositivo., en este caso el 55222. Dicha información, el código de confirmación 401 vuelve a ir cofrada con la clave entre pasarelas.

A continuación en la figura 37 se observa la confirmación de migración pero entre PM y el dispositivo. Además del código de confirmación 401, dicha traza contiene la información para conectarse la pasarela e iniciar la comunicación segura, es decir, Ip, puerto y clave, en este caso:

Confirmación Ip PE - 192.168.1.4

Puerto - 7777

Clave - Gateway222222222

Una vez se ha procedido al intercambio de datos, el dispositivo debe enviar una solicitud de conexión a PE y este, al confirmar dicha solicitud, reconfigura los parámetros de identidad (ID y Num Sec) para el continuar con el envío de datos. De esta manera configuramos de nuevo el secreto compartido entre dispositivos y aseguramos la autenticación de dispositivos.

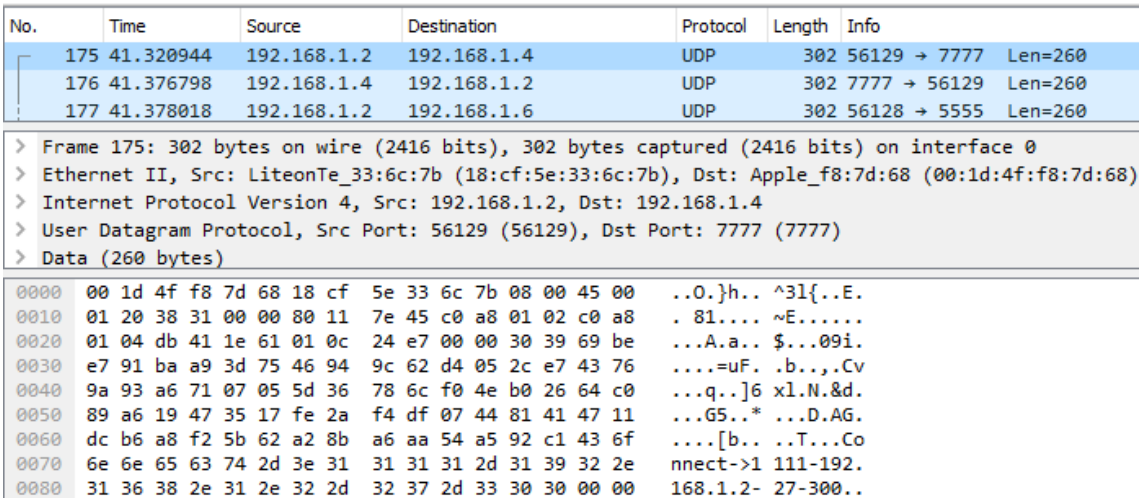

Figura 38. Captura de Wireshark: Solicitud de Conexión (código 300) con los datos antiguos de identidad para autenticarse en PE, traza entre el Dispositivo y PE.

En la figura 38 se observa la traza de solicitud de conexión entre Dispositivo y PE. Datos cifrados con la nueva clave que se le ha proporcionado al dispositivo, contiene los datos de autenticación que se le han proporcionado a PE. Como se observa en la figura, al cambiar de pasarela, el dispositivo ha abierto un nuevo puerto de comunicaciones (56129) y ahora envía datos al puerto de PE (7777).

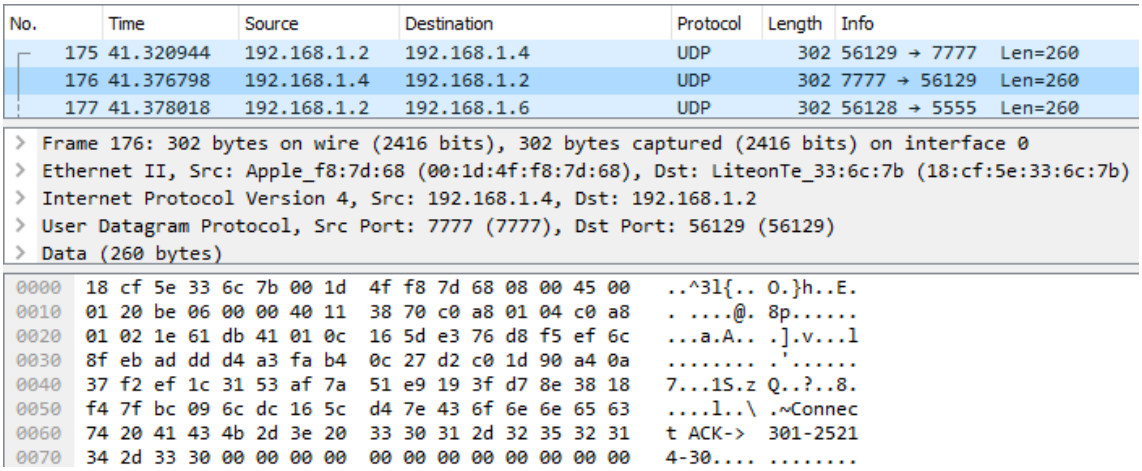

Figura 39. Captura de Wireshark: ACK de Conexión (código 301) con los nuevos datos de identidad para el Dispositivo, desde PE al Dispositivo.

En la figura anterior, tras confirmar que el dispositivo es quien dice ser pues cumple los requisitos de identidad al cifrar correctamente los parámetros de identidad que se le han suministrado a PE desde PM, PE envía la confirmación de conexión 301 con los nuevos datos de identidad ID y Num Sec. Dichos parámetros son los que debe utilizar el dispositivo a partir de ese instante para que sus datos sean validos. Esta traza irá cifrada con la clave que se le ha proporcionado a PE para comunicarse con el dispositivo e ira envía al puerto del dispositivo desde el cual se ha enviado la solicitud de conexión.

Tras le recepción de este mensaje, el dispositivo reconfigura sus parámetros de identidad, como se observa en la figura, nuevo Id = 25214 y nuevo Num Sec = 30. A continuación procede

a desconectarse de la antigua pasarela y a enviar datos por la nueva conexión segura creada. El envío de datos es idéntico al descrito en el apartado 5.2 con el cambio de los parámetros de identidad y clave de cifrado-descifrado.

5.4 Desconexión

Como se ha explicado en el apartado anterior, en caso de migración, o en caso de desconexión Este mensaje no tiene porque ser *mandatory* pues las tareas de mantenimiento deberán mantener al día el repositorio de dispositivos activos en cada pasarela. Para prevenir de la desconexión sin autorización de cualquier dispositivo, esta solicitud irá acompañada de los datos de identidad. Cabe recordar que no requiere ACK pues el dispositivo ya habrá cerrado los recursos asociados a dicha conexión.

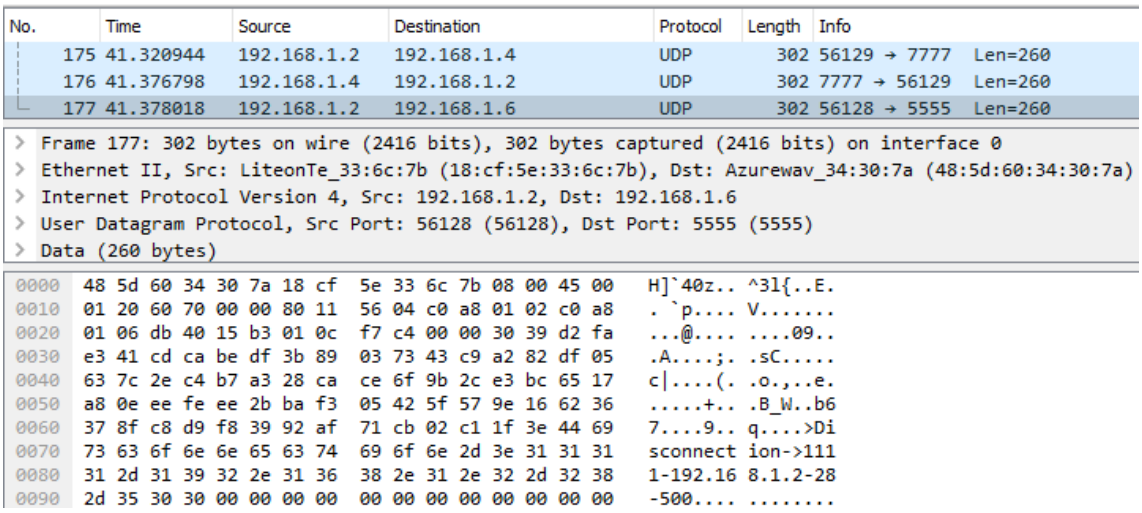

Figura 40. Captura de Wireshark: Solicitud de Desconexión (código 500) con los datos identidad para autenticarse en la pasarela correspondiente del Dispositivo a la pasarela.

En la figura se observa como el mensaje es enviado por el puerto abierto desde el dispositivo para comunicarse con PM (56128). Es la última traza enviado por ese puerto, pues la nueva conexión con PE ha sido abierta en el puerto 55222. Tras el envío de esta traza, los recursos asignados a la conexión con PM con desactivados.

5.5 Reconexión

En caso de pérdida de señal, un dispositivo puede reconectarse a una pasarela por medio de una Solicitud de Conexión (código 300). El procedimiento es idéntico al proceso de conexión tras una migración explicado en el apartado 5.3. Si la solicitud es válida, en la respuesta por parte de la pasarela se reconfiguraran los parámetros de identidad y se podrá proceder al envío de datos.

5.6 Diagrama Completo de Funcionamiento

Una vez instalados todos los elementos de nuestra red, la comunicación entre ellos es la siguiente:

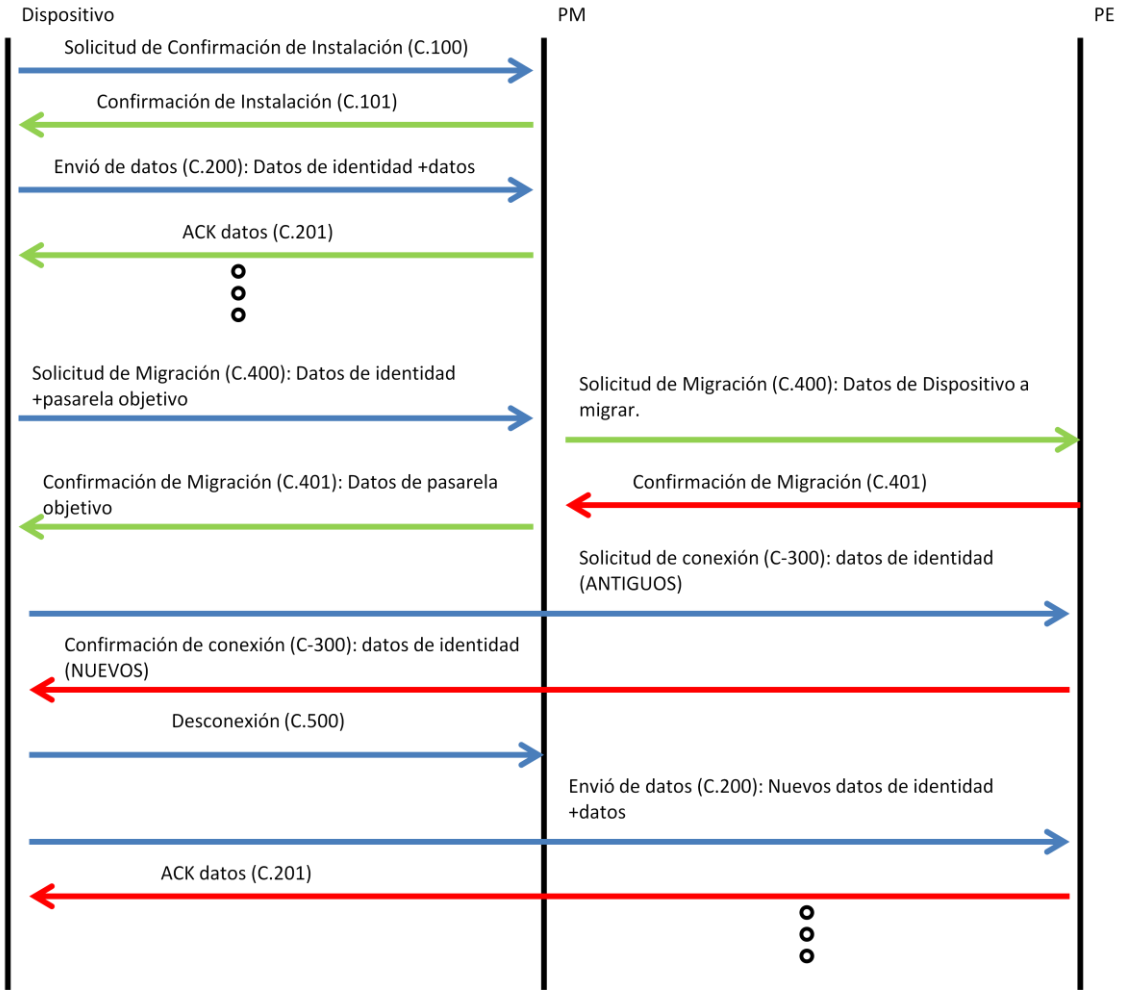

Figura 41. Diagrama Completo de Comunicación.

La figura 41 muestra el comportamiento global del sistema una vez instalado y desplegado en el que cada color representa un cifrado con una clave distinta:

i) Azul – Clave del Dispositivo
ii) Verde – Clave de PM
iii) Rojo – Clave de PE

De esta manera, en ningún momento circula por la red información en plano que pueda comprometer a los dispositivos y, de esta manera, se consigue un entorno seguro ante usuarios maliciosos. Durante el funcionamiento del dispositivo, el proceso explicado en la figura 41, se producirá cada vez que el dispositivo migre entre pasarelas.

Capítulo 6. Conclusiones y Líneas Futuras

6.1 Conclusiones

El objetivo de este Trabajo de Fin de Grado era desarrollar un entorno IoT seguro en la que una serie dispositivos móviles pudiera enviar datos y migrar entre pasarelas de manera segura. Se ha intentado proceder siguiendo una estructura definida en el proceso de desarrollo de software expuesto en los primero apartados del TFG. Una gran parte del tiempo de ha ocupado en el proceso de investigación de herramienta de securizacion adaptables a este tipo de entonarnos y sus limitaciones. Gran parte de las conclusiones se han obtenido de la etapa de investigación.

La primera gran conclusión ha sido la falta total de estandarización. La información a la que se tiene acceso se limita a una serie de recomendaciones pero sin definir una estructura uniforme de las herramientas a utilizar para securizar este tipo de entornos. En la mayoría de los casos se opta por una securizacion débil o el uso de TLS/SSL sobre el protocolo IoT para garantizar la seguridad. En vitas de esta falta de estandarización, me he visto obligado a buscar soluciones globales para los protocoles existentes, como le cifrado mediante AES 128, y soluciones también adaptadas a las limitación de los dispositivos *edge* de las rede, como puede haber sido la elección de UDP para el transporte de tramas.

Dotar de indepencia a una red sin la necesidad de la intervención humana ha sido un punto crítico. Envista de las limitaciones computacionales de los dispositivos, los complejos algoritmos de negociación de claves o la utilización de certificados tenía que quedar descartada. Debido a esto, para dotar de independencia a la red se propone gastar cierto tiempo en el proceso de instalación y despliegue de la red. Haciendo esto y mediante el sistema de distribución de datos que se propone, en l que ningún mensaje aparece en claro por la red, se consigue que la red sea independientes tras un periodo de intervención humana durante el proceso de despliegue e instalación. La independencia va ligada al grado de seguridad de la red, pues al no trasmitirse datos en claro, los ataque por fuerza bruta con claves AES 128 no son una posibilidad. Ataque MITM (Man In The Middle) también quedan descartados, pues un usuario malicioso a la escucha no puede leer ningún dato que se tramite. Se intenta que la red solo se vea afectada en caso de que un elemento de la misma se avería físicamente, que ningún tipo de agente externo pueda afectar a la misma por medio del envío de órdenes o datos maliciosos suplementando alguna identidad, a su vez manteniendo la total confidencialidad de los datos transmitidos. Las tareas de mantenimiento pretenden activar señales para identificar estos fallos en la red en elementos cruciales como son las pasarelas. Las tareas de verificación de dispositivos se deberían definir en el servidor de recepción los datos. Definiendo una ventana de tiempo en la que cierto dispositivo debe enviar información, en el entorno no se he observado una manera eficiente de verificar la salud de los dispositivos pues todas las pasarelas no pueden contemplar todos los dispositivos y verificar si han transmitido a través de otra pasarela de la red. No es eficiente de cara a redes grandes sin inundar la red. La intervención humana es necesaria pero pasado el punto de instalación y despliegue se pretende que sea mínima.

Aun dotando de segura al transporte de información y migración en el *edge* de red IoT, para garantizar la seguridad de la red, habría que combinar la funcionalidad desarrollada con otras herramientas de seguridad como sistemas de detección de intrusos o la utilización de software actualizado. Además de este tipo de herramientas, se deberán construir claves robustas y los usuarios que tengan acceso a los dispositivos para las operaciones de mantenimiento, deberán construir usuarios con contraseñas asociadas de un alto grado de seguridad. En definitiva, seguir una política de seguridad adecuada para la salud de la red.

Se puede concluir, además por el caso de uso desarrollado, que el objetivo principal de la funcionalidad se ha cumplido. Se he conseguido que en un entorno IoT, en la que existen claras limitaciones de recursos computacionales, se transmita información de manera segura, que lo dispositivos móviles migren entre pasarelas de manera segura y que un usuario externo, desconociendo claves y secretos compartidos NO pueda entender cualquier dato transmitido.

A título personal, desarrollar una funcionalidad aplicable dispositivos y entornos reales ha sido de gran utilidad. Estar envuelto en todas las etapas de desarrollo de un sistema y afrontar los retos encontrados en cada etapa ha sido un reto tremendamente constructivo. Por último decir que desarrollar una aplicación desde 0, en la que todo el proceso ha sido fruto de tu trabajo ha sido una satisfacción.

6.2 Líneas Futuras

En primer paso tras desarrollar este proyecto sería trasladar a la realidad el caso de uso desarrolla en este TFG. La implantación del software desarrollado en dispositivos físicos y analizar el comportamiento, es decir, verificar que se cumple el comportamiento visto en dispositivos virtuales. Además de verificar el comportamiento respecto a criptografía y seguridad, habría que analizar el impacto en el consumo energético que supondría la aplicación de las herramientas de seguridad desarrolladas en este proyecto en dispositivos físicos. Analizar el consumo energético de los dispositivos físicos con y sin la funcionalidad desarrollada.

En segundo lugar y también contemplado en este proyecto, sería el desarrollo del entorno local en la que la que aparece un nuevo elemento, el Servidor de Autenticación. Con el trabajo realizado hasta el momento como base, cabria desarrollar el nuevo dispositivo sobre lo ya desarrollado, siguiendo el modelo de desarrollo incremental.

Además de las herramientas de autenticación y seguridad, similares a un desafío-respuesta, habría que contemplar la posibilidad de sustituir el número de secuencia utilizado para sincronizar el flujo de datos por una marca de tiempo o simplemente añadir este parámetro a la traza. Dicha marca de tiempo estaría sincronizada por medio de la introducción de un servidor de tiempos dentro de nuestra red, ya existen sendos protocolos para la sincronización de dispositivos, habría que variar las trazas expuestas en este proyecto para sustituir el número de secuencia por una marca temporal o para añadir dicha marca. Este paso dotaría de un mayor grado se seguridad al sistema, pero habría que observar el impacto en la utilización de recursos con la introducción de esta funcionalidad.

En un futuro habría que contemplar una serie de pruebas más intensivas y exigentes para verificar el comportamiento de los dispositivos. Entre las pruebas y procedimientos futuros se deberían realizar pruebas de carga y estrés en la que interactuasen una cantidad elevada de dispositivos. Se deberían realizar pruebas en la que procesos de migración, envío de datos e instalación estén ocurriendo alrededor de la red de manera simultánea. De esta manera se podría definir correctamente los recursos necesarios para las pasarelas y servidor de autenticación.

Además de las pruebas mencionadas, vivimos en un mundo en el que la tecnología sufre cambios constantes. Por esta razón habría que desarrollar un sistema para poder adaptar mejorar a entornos ya existentes y en funcionamiento, por ejemplo, por medio de la aplicación de parches o *upgrades* de software. Dichos cambios o mejoras podrían afectar a cambios en el algoritmo de cifrado o cambios en el sistema de autenticación, secreto compartido, método de autenticación etc. Dotar de mayor independencia al sistema para que se mantenga al margen de la gestión de un administrador.

Por último, una vez probados y mejorados entornos de menor tamaño, habría que analizar la posibilidad de adoptar las soluciones más pequeñas a entonos grandes, entornos como ciudades, zonas rúales de gran extensión, etc.

6.3 Bibliografía

[1] **Laboratorio Nacional de Calidad del Software** Ingeniería del software: metodologías y ciclos de vida. - Julio de 2015

[2] **Palau Salvador, Carlos Enrique** Arquitectura de Seguridad Cliente/Servidor // Seguridad. Valencia: Departamento de Comunicaciones. ETSIT UPV, 2014.

[3] **Designing the Internet of Things** [En Linea] // Micrium – Agosto de 2016 - https://www.micrium.com/iot/internet-protocols/

[4] **Análisis de IoT** [En Línea] // Intel – Agosto de 2016 - http://www.intel.es/content/www/es/es/internet-of-things/overview.html

[5] **11 Intenet of Things Protocols You Need to Know About** [En Linea] // RS Components – Agosto de 2016 -

http://www.rs-online.com/designspark/electronics/eng/knowledge-item/eleven-internet-of-things-iot-protocols-you-need-to-know-about

[6] Video - **6LoWPAN Tutorial – A Wireless Extension of the Internet** [En Linea] // Texas Instruments – Agosto de 2016 -

https://www.youtube.com/watch?v=zzoZNG_NB_c -6LoWPAN Tutorial – A Wireless Extension of the Internet

[7] **Cristina Peña y Carlos Ralli** IPv6 Motor Internet de las cosas IoT [En Línea] // Blogthnikbig -Telefónica – Agosto de 2016 -

http://blogthinkbig.com/ipv6-motor-internet-de-las-cosas-iot/

[8] **ZigBee** [En Línea] // ZigBee Alliance – Agosto de 2016 -

http://www.zigbee.org/

[9] **RFC 7252 Constrained Application Protocol** [En Linea] // CoAP Technology - Agosto de 2016 -

http://coap.technology/

[10] **MQTT** [En Línea] // MQTT Org – Agosto de 2016 -

http://mqtt.org/

[11] **Security in 802.15.4 and ZigBee networks** [En Linea] // Libelium – Agosto de 2016 - http://www.libelium.com/security-802-15-4-zigbee/

[12] **Transport Layer Security** [En Línea] // Wikipedia – Agosto de 2016 - https://es.wikipedia.org/wiki/Transport_Layer_Security,

[13] **Transport Layer Security** [En Línea] // Wikipedia – Agosto de 2016 - https://en.wikipedia.org/wiki/Transport_Layer_Security